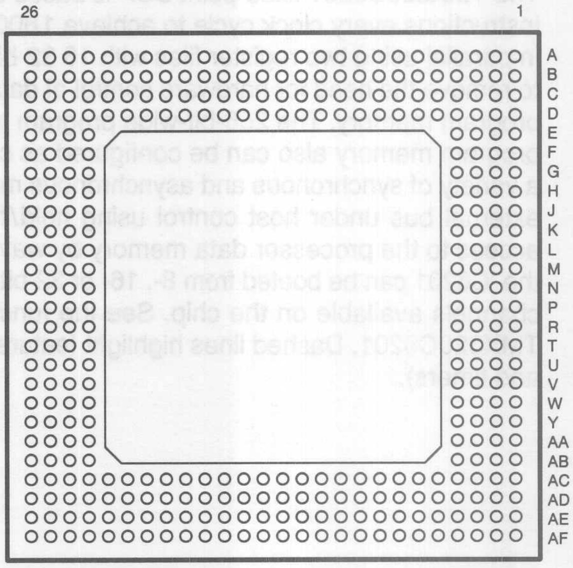
TMX320C6201
DIGITAL SIGNAL PROCESSOR

SPRS051 – JANUARY 1997

- **Highest Performance Fixed-Point Digital Signal Processor (DSP):**
  - **1600 MIPS @ 200 MHz**
  - **Five-ns Cycle Time**
  - **Eight × 32-Bit Instructions/Cycle**
- **VelociTI™ Advanced Very-Long Instruction Word (VLIW) Architecture:**
  - **Eight Independent Functional Units**
  - **Two 16-Bit Multipliers (32-Bit Results)**
  - **Six Arithmetic Logic Units (ALUs) (32-/40-Bit)**
  - **Load/Store Architecture: 32 32-Bit General-Purpose Registers**
  - **Instruction Packing to Reduce Code Size**
  - **100% Conditional Instructions**
- **Instruction Set Features**
  - **Byte-Addressable 8-, 16-, 32-Bit Data**
  - **32-Bit Address Range**
  - **Dual-Endian Support**
  - **8-Bit Headroom Available**
  - **Saturation**
  - **Bit-Field Extract, Set, Clear**
  - **Bit Counting**
  - **Normalization**
  - **Register or 5-Bit Constant Index**
  - **15-Bit Positive/Negative Offset From Either B14 or B15 Register**
- **1 M-Bit On-Chip SRAM**
  - **512K-Bit Internal-Program/Cache Memory**
  - **512K-Bit Internal-Data Memory**
- **32-Bit External Memory Interface (EMIF)**
  - **Glueless Interface To Synchronous Memories:**
    Synchronous Dynamic Random-Access Memory (SDRAM)
    Synchronous Burst Static RAM (SBSRAM)
  - **Glueless Interface to Asynchronous Memories:**
    Static RAM (SRAM)
    Erasable Programmable Read-Only Memory (EPROM)
  - **Glueless Interface to External Controller Devices:**
    First-In/First-Out (FIFO)

GGP 352-PIN BGA PACKAGE
(TOP VIEW)



- **Two-Channel Boot-Loading Direct Memory Access (DMA) Coprocessor**
- **16-Bit Host-Access Port**
- **Flexible Phase-Locked Loop (PLL) Clock Generator (×1, ×2, ×4)**
- **IEEE Std. 1149.1 (JTAG†) Scan-Based Emulation**
- **352-Pin Ball Grid Array (BGA) Package**
- **0.25 micron / Five-Level Metal Process**
- **0°C–90°C Operating Case Temperature Range**

PRODUCT PREVIEW

Please be aware that an important notice concerning availability, standard warranty, and use in critical applications of Texas Instruments semiconductor products and disclaimers thereto appears at the end of this data sheet.

---

† IEEE Standard 1149.1-1990, IEEE Standard Test Access Port and Boundary-Scan Architecture

VelociTI is a trademark of Texas Instruments.

PRODUCT PREVIEW information concerns products in the formative or design phase of development. Characteristic data and other specifications are design goals. Texas Instruments reserves the right to change or discontinue these products without notice.

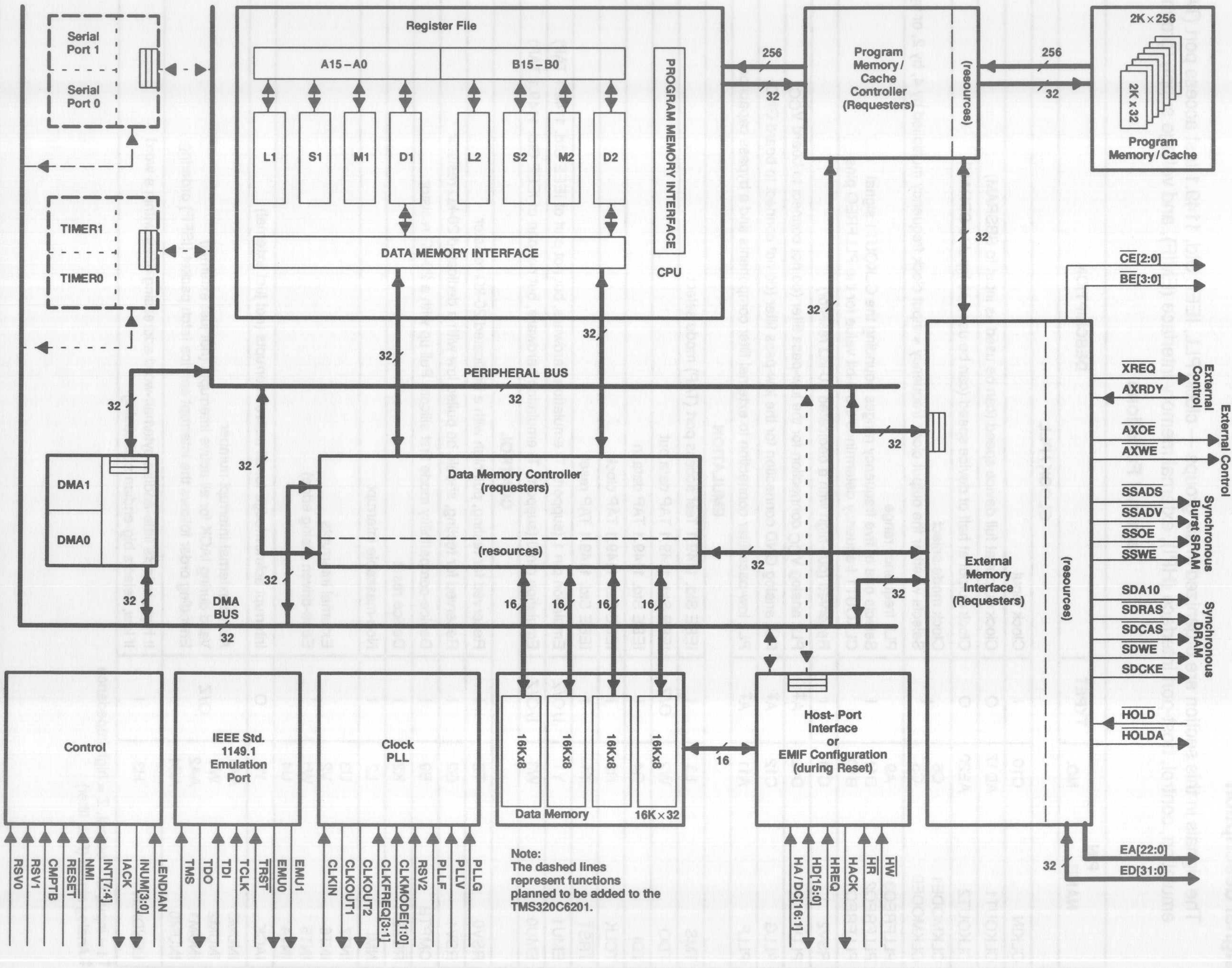
Copyright © 1997, Texas Instruments Incorporated

description

The TMX320C6201 fixed-point DSP is based on the VelociTI™ VLIW 'C6200 CPU core. It executes up to eight instructions every clock cycle to achieve 1600 MIPS performance at 200 MHz. Eight functional units operate in parallel using two register files with 16 32-bit registers each. Program parallelism is defined at compile time to remove the need for hardware control of data interlocking. The 1M-bit on-chip RAM is split between data and program memory. The 256-bit-wide program memory fetches eight 32-bit instructions every single cycle. The program memory also can be configured as cache memory. The external memory interface (EMIF) supports a variety of synchronous and asynchronous memories including SBSRAM and SDRAM. EMIF can release the external bus under host control using hold/hold-acknowledge protocol. Additionally, the host has a direct access to the processor data memory by way of the separate 16-bit host-port interface. Following power up, the 'C6201 can be booted from 8-, 16- or 32-bit external ROM using one of the two direct memory access (DMA) channels available on the chip. See the functional block diagram for a high-level functional diagram of the TMX320C6201. Dashed lines highlight features that are to be added in the next release (including serial ports and timers).

PRODUCT PREVIEW

functional block diagram



Note:
The dashed lines represent functions planned to be added to the TMS320C6201.

PRODUCT PREVIEW

PRODUCT PREVIEW

signal description

The signals in this section are organized in six groups — clock/PLL, IEEE Std. 1149.1 test access port (TAP) emulation, control, host-port interface (HPI), external memory interface (EMIF), and voltage supply/ground.

Terminal Functions

PIN NAME	PIN NO.	TYPE†	DESCRIPTION
			CLOCK / PLL
CLKIN	C10	I	Clock Input
CLKOUT1	AD17	O	Clock output at full device speed (can be used as input to SBSRAM)
CLKOUT2	AE20	O	Clock output at half of device speed (can be used as input to SDRAM)
CLKMODE1 CLKMODE0	C6 C5	I	Clock mode select Selects whether the output clock frequency = input clock frequency multiplied by 4, by 2, or by 1
PLLFREQ3 PLLFREQ2 PLLFREQ1	A9 D11 B10	I	PLL frequency range Selects one of five frequency ranges bounding the CLKOUT1 signal CLKOUT1 frequency determines the 3-bit value for the PLLFREQ pins
RSV2	C11	I	Reserved (pull-high with a dedicated 20-kΩ resistor)
PLLV	D12	A‡	PLL analog V_{CC} connection for the low-pass filter (do not connect to board V_{CC})
PLLG	C12	A‡	PLL analog GND connection for the low-pass filter (do not connect to board GND)
PLLF	A11	A‡	PLL low-pass filter connection to external filter components and a bypass capacitor
			EMULATION
TMS	L3	I	IEEE Std. 1149.1 Test access port (TAP) mode select
TDO	W2	O/Z	IEEE Std. 1149.1 TAP data out
TDI	R4	I	IEEE Std. 1149.1 TAP data in
TCLK	R3	I	IEEE Std. 1149.1 TAP clock
$\overline{\text{TRST}}$	T1	I	IEEE Std. 1149.1 TAP reset
EMU1	Y1	I/O/Z	Emulation pin 1 (supports TI emulation hardware, but not part of IEEE Std. 1149.1 TAP)
EMU0	W3	I/O/Z	Emulation pin 0 (supports TI emulation hardware, but not part of IEEE Std. 1149.1 TAP)
			CONTROL
RSV0	T2	I	Reserved for testing, pull high with a dedicated 20-kΩ resistor
RSV1	G2	I	Reserved for testing, should be pulled low with a dedicated 20-kΩ resistor
CMPTB	B9	I	Device-compatibility mode first silicon. Pull up with a 20-kΩ resistor
$\overline{\text{RESET}}$	K2	I	Device reset
NMI	L2	I	Non-maskable interrupt
INT7 INT6 INT5 INT4	U3 V2 W1 U4	I	External interrupts Edge-driven (rising edge)
IACK	Y2	O	Interrupt acknowledge for all active interrupts (not just external)
INUM3 INUM2 INUM1 INUM0	AA1 W4 AA2 AB1	O/Z	Active external-interrupt number Valid during IACK for all active interrupts (not just external) Encoding order follows the interrupt service fetch packet (ISFP) ordering
LENDIAN	H3	I	If high, selects little-endian byte/half-word addressing order within a word If low, selects big-endian addressing

† I = input, O = output, Z = high impedance
‡ Analog signal (PLL filter)

Terminal Functions (Continued)

PIN NAME	PIN NO.	TYPE†	DESCRIPTION
HOST PORT INTERFACE			
HA/DC16	E25	I	Address from host/device configuration Host address valid during $\overline{\mathrm{RESET}}$ high (not active) Device configuration valid during $\overline{\mathrm{RESET}}$ low (active)
HA/DC15	D26		
HA/DC14	F23		
HA/DC13	D25		
HA/DC12	C26		
HA/DC11	E23		
HA/DC10	D24		
HA/DC9	C23		
HA/DC8	D22		
HA/DC7	A24		
HA/DC6	B23		
HA/DC5	C22		
HA/DC4	D21		
HA/DC3	A23		
HA/DC2	B22		
HA/DC1	C21		
HD15	B13	I/O/Z	Host-port data
HD14	B14		
HD13	C14		
HD12	B15		
HD11	D15		
HD10	B16		
HD9	A17		
HD8	B17		
HD7	D16		
HD6	B18		
HD5	A19		
HD4	C18		
HD3	B19		
HD2	C19		
HD1	B20		
HD0	B21		
HREQ	D10	I	Request from host
HACK	B8	O	Request acknowledge to host
$\overline{\mathrm{HR}}$	C9	I	Read strobe from host
$\overline{\mathrm{HW}}$	A8	I	Write strobe from host
EMIF (CONTROL SIGNALS COMMON TO ALL TYPES OF MEMORY)			
$\overline{\mathrm{CE2}}$	AD26	O/Z	Memory-space enables Enabled by bits 24 and 25 of the byte address Only one asserted during any external-data access
$\overline{\mathrm{CE1}}$	AB24		
$\overline{\mathrm{CE0}}$	AC26		
$\overline{\mathrm{BE3}}$	AB25	O/Z	Byte-enable control (decoded from the two lowest bits of the internal address) Byte-write enables for most types of memory Can be directly connected to SDRAM read- and write-mask signal (SDQM)
$\overline{\mathrm{BE2}}$	AA24		
$\overline{\mathrm{BE1}}$	Y23		
$\overline{\mathrm{BE0}}$	AA26		

† I = input, O = output, Z = high impedance

Terminal Functions (Continued)

PIN NAME	PIN NO.	TYPE†	DESCRIPTION
EMIF (ADDRESS)			
EA22	J24	O/Z	External address (word address)
EA21	H26		
EA20	K24		
EA19	J26		
EA18	K25		
EA17	L24		
EA16	K26		
EA15	M26		
EA14	M25		
EA13	P25		
EA12	P24		
EA11	R25		
EA10	T26		
EA9	R23		
EA8	U26		
EA7	U25		
EA6	T23		
EA5	V26		
EA4	V25		
EA3	W26		
EA2	V24		
EA1	W25		
EA0	Y26		
EMIF (DATA)			
ED31	AB2	I/O/Z	External data
ED30	AC1		
ED29	AA4		
ED28	AD1		
ED27	AC3		
ED26	AD4		
ED25	AF3		
ED24	AE4		
ED23	AD5		
ED22	AF4		
ED21	AE5		
ED20	AD6		
ED19	AE6		
ED18	AD7		
ED17	AC8		
ED16	AF7		
ED15	AD9		
ED14	AD10		
ED13	AF9		
ED12	AC11		
ED11	AE10		
ED10	AE11		
ED9	AF11		
ED8	AE14		
ED7	AF15		
ED6	AE15		
ED5	AF16		
ED4	AC15		

† I = input, O = output, Z = high impedance

Terminal Functions (Continued)

PIN NAME	PIN NO.	TYPE†	DESCRIPTION
EMIF (DATA) (CONTINUED)			
ED3	AE17	I/O/Z	External data
ED2	AF18		
ED1	AF19		
ED0	AC17		
EMIF (EXTERNAL CONTROL)			
$\overline{\text{XREQ}}$	Y24	O/Z	External controller request
AXRDY	W23	I	External controller read
EMIF (ASYNCHRONOUS-SRAM/EXTERNAL CONTROL)			
$\overline{\text{AXOE}}$	AC24	O/Z	Asynchronous SRAM/external controller output enable
$\overline{\text{AXWE}}$	AD23	O/Z	Asynchronous SRAM/external controller write enable
EMIF (SYNCHRONOUS-BURST SRAM CONTROL)			
$\overline{\text{SSADS}}$	AC20	O/Z	SBSRAM address strobe
$\overline{\text{SSADV}}$	AF22	O/Z	SBSRAM address-advance strobe
$\overline{\text{SSOE}}$	AF21	O/Z	SBSRAM output enable
$\overline{\text{SSWE}}$	AD19	O/Z	SBSRAM write enable
EMIF (SYNCHRONOUS DRAM CONTROL)			
SDA10	AD21	O/Z	SDRAM address 10 (separate for refresh)
$\overline{\text{SDRAS}}$	AF24	O/Z	SDRAM row address strobe
$\overline{\text{SDCAS}}$	AD22	O/Z	SDRAM column-address strobe
$\overline{\text{SDWE}}$	AF23	O/Z	SDRAM write enable
SDCKE	AE22	O/Z	SDRAM clock enable
EMIF (BUS ARBITRATION)			
$\overline{\text{HOLD}}$	AA25	I	Hold request from the host
$\overline{\text{HOLDA}}$	AF20	O	Hold request acknowledge to the host

† I = input, O = output, Z = high impedance

PRODUCT PREVIEW

Terminal Functions (Continued)

PIN NAME	PIN NO.	TYPE†	DESCRIPTION
3.3 V SUPPLY VOLTAGE			
V_{DD3V}‡	B7	I	Supply voltage
	A10		
	A15		
	A18		
	D17		
	A21		
	A22		
	G24		
	G25		
	H25		
	J25		
	L25		
	N23		
	R26		
	T24		
	U24		
	W24		
	AB26		
	AC25		
	AC22		
	AC21		
	AE21		
	AC19		
	AD18		
	AE18		
	AF17		
	AD15		
	AD13		
	AD11		
	AC10		
	AF6		
	AF5		
	AC6		
	AB4		
	AB3		
	Y4		
	N3		
	M3		
	F3		
	C1		

† I = input, O = output, Z = high impedance

‡ PLLV and PLLG signals are not part of the external voltage supply or ground and must not be connected to any of the V_{DD3V}, V_{DD2V}, or GND pins.

Terminal Functions (Continued)

PIN		TYPE†	DESCRIPTION
NAME	NO.		
2.5 V SUPPLY VOLTAGE			
V_{DD2V}‡	D6	I	Supply voltage
	D7		
	A5		
	B6		
	B11		
	B12		
	A12		
	C15		
	A16		
	A20		
	C20		
	D20		
	K23		
	M24		
	N25		
	AD14		
	AF12		
	AE12		
	AC12		
	AE8		
	AD8		
	T4		
	T3		
	U1		
	P2		
	M1		
	K1		
	F1		
	H4		
	E1		
	B2		
	C3		
	D4		
	AE2		
	AD3		
	AC4		
	D23		
	C24		
	B25		
	AC23		
	AD24		
	AE25		

† I = input, O = output, Z = high impedance

‡ PLLV and PLLG signals are not part of the external voltage supply or ground and must not be connected to any of the V_{DD3V}, V_{DD2V}, or GND pins.

PRODUCT PREVIEW

Terminal Functions (Continued)

PIN NAME	PIN NO.	TYPE†	DESCRIPTION
2.5 V SUPPLY VOLTAGE (CONTINUED)			
V_{DD2V}‡	D9 D14 D18 J23 P23 V23 J4 N4 V4 AC9 AC13 AC18	I	Supply voltage
GROUND			
GND‡	A4 B5 C7 D13 C13 C16 C17 D19 E24 F24 G26 L23 M23 L26 N24 R24 T25 U23 Y25 AA23 AB23 AE23 AD20 AE19 AC16 AD16 AE16 AC14 AE13 AD12 AF10	I	Ground

† I = input, O = output, Z = high impedance

‡ PLLV and PLLG signals are not part of the external voltage supply or ground and must not be connected to any of the V_{DD3V}, V_{DD2V}, or GND pins.

Terminal Functions (Continued)

PIN NAME	NO.	TYPE†	DESCRIPTION
			GROUND (CONTINUED)
	AE9		
	AF8		
	AE7		
	AC7		
	AC5		
	AC2		
	AA3		
	Y3		
	V3		
	V1		
	U2		
	N2		
	J3		
	G3		
	F2		
	G4		
	E3		
	A1		
	B1		
	A2		
	C2		
	B3		
GND‡	AE1	I	Ground
	AF1		
	AD2		
	AF2		
	AE3		
	A25		
	A26		
	B26		
	C25		
	B24		
	AF25		
	AF26		
	AE24		
	AE26		
	AD25		
	A13		
	A14		
	AF13		
	AF14		
	N1		
	P1		
	N26		
	P26		

† I = input, O = output, Z = high impedance

‡ PLLV and PLLG signals are not part of the external voltage supply or ground and must not be connected to any of the V_{DD3V}, V_{DD2V}, or GND pins.

PRODUCT PREVIEW

CPU architecture

The 'C6200 CPU is the central building block of all members of the '320C62xx family of digital signal processors (DSPs), including the '320C6201.

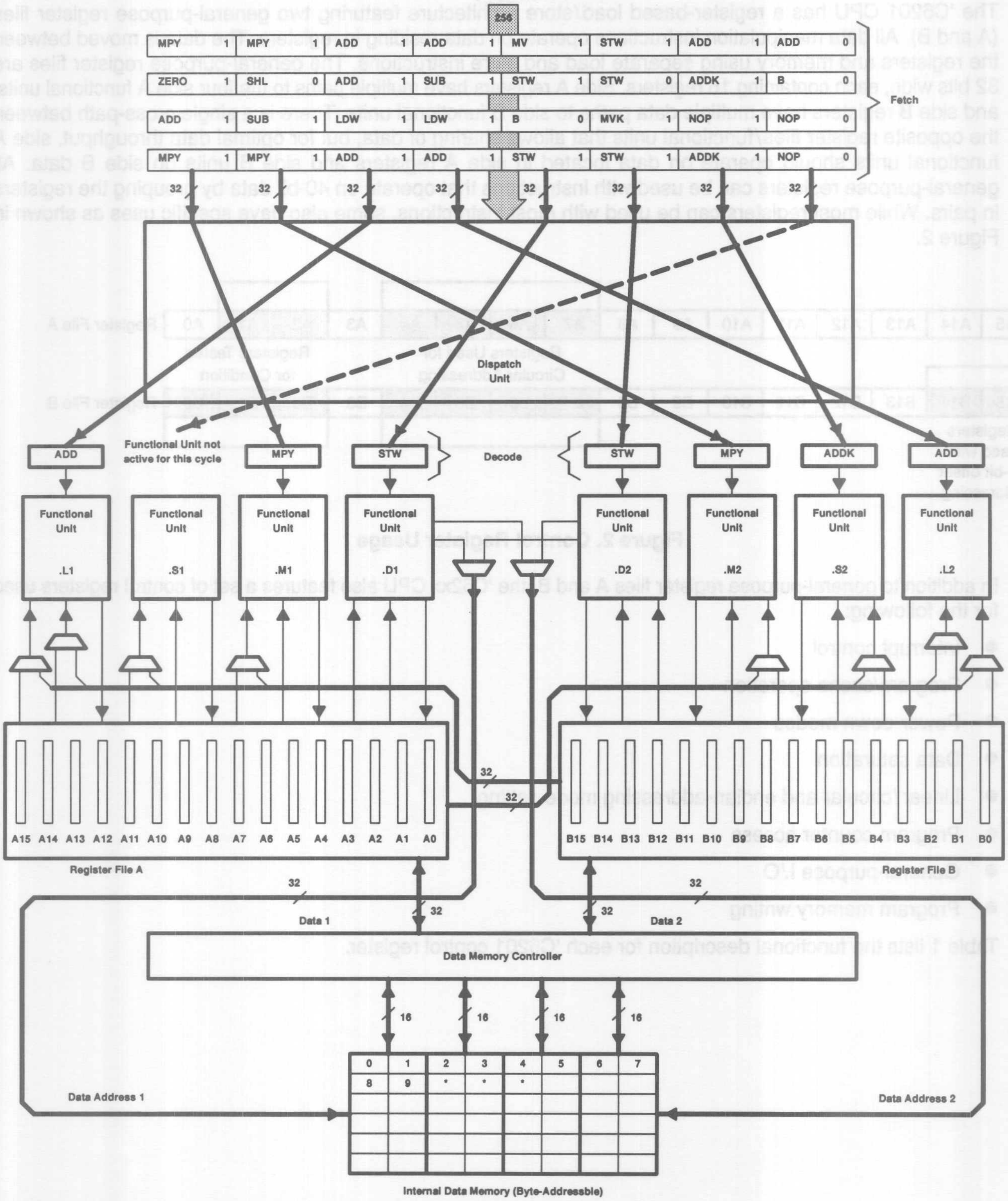
The CPU uses very-long instruction words (256-bits wide) to supply up to eight 32-bit instructions to the eight functional units during every clock cycle. Fetch packets are always 256-bits wide; however, the execute packets can vary in size as shown in Figure 1. The variable-length execute packets are a key memory-saving feature, distinguishing the 'C6200 CPU from other VLIW architectures.

The CPU features two sets of functional units. Each set contains four units and a register file. The two register files contain 16 32-bit registers each for the total of 32 general-purpose registers. The two sets of functional units, along with two register files, comprise sides A and B of the CPU (see Figure 1). The four functional units on each side of the CPU can share the 16 registers belonging to that side. Additionally, each side features a single data bus connected to all registers on the other side, by which the two sets of functional units can cross-exchange data from the register files on opposite sides. While register access by functional units on the same side of the CPU as the register file can service all the units in a single clock cycle, register access using the register file across the CPU supports only one read and one write per cycle.

Another key feature of the 'C6200 CPU is the load/store architecture, where all instructions operate on registers (as opposed to data in memory). Two sets of data addressing units (.D1 and .D2) are exclusively responsible for all data transfers between the register files and the memory. The data address driven by the .D units allows data addresses generated from one register file to be used in data loads and stores affecting the other register file. The 'C6200 CPU supports a variety of indirect-addressing modes using linear- or circular-addressing modes with 5- or 15-bit offsets. All instructions are conditional, and most instructions can access any one of the 32 registers. Some registers are also singled out to support specific addressing or to hold the condition for conditional instructions (if the condition is not automatically "true"). The two .M functional units are dedicated to multiplies. The two .S and .L functional units perform general arithmetic, logical, and branch functions with results available at the rate of every clock cycle (the latency can vary between one and five cycles due to the multi-stage execution pipeline, with most instructions executing in one cycle).

The VelociTI VLIW processing flow begins when a 256-bit-wide instruction fetch packet (IFP) is fetched from the internal program memory (that also can be configured as cache). The 32-bit instructions destined for the individual functional units are linked together by the bit in the least significant bit (LSB) position of the instructions. The instructions that are linked together for simultaneous execution (up to eight in total) comprise an execute packet. A 0 in the LSB of an instruction breaks the chain, effectively placing the instructions that follow it in the next execute packet. If an execute packet crosses the fetch packet boundary (256 bits wide), the compiler places it in the next fetch packet, while the remainder of the current fetch packet is padded with NOP instructions. The number of execute packets within a fetch packet can vary from one to eight. Execute packets are dispatched to their respective functional units at the rate of one per clock cycle (see Figure 1) and the next 256-bit fetch packet is not fetched until all the execute packets from the current fetch packet have been dispatched. After decoding, the instructions simultaneously drive all active functional units for a maximum execution rate of eight instructions every clock cycle. While most results are stored in 32-bit registers, they can be stored in memory as bytes or half-words as well, effectively making all stores and loads byte-addressable for savings in memory requirements.

CPU architecture (continued)


256
MPY 1 MPY 1 ADD 1 ADD 1 MV 1 STW 1 ADD 1 ADD 0
ZERO 1 SHL 0 ADD 1 SUB 1 STW 1 STW 0 ADDK 1 B 0
ADD 1 SUB 1 LDW 1 LDW 1 B 0 MVK 1 NOP 1 NOP 0
MPY 1 MPY 1 ADD 1 ADD 1 STW 1 STW 1 ADDK 1 NOP 0
Fetch
32
32
32
32
32
32
32
32
Dispatch Unit
ADD
Functional Unit not active for this cycle
MPY
STW
Decode
STW
MPY
ADDK
ADD
Functional Unit .L1
Functional Unit .S1
Functional Unit .M1
Functional Unit .D1
Functional Unit .D2
Functional Unit .M2
Functional Unit .S2
Functional Unit .L2
32
32
A15 A14 A13 A12 A11 A10 A9 A8 A7 A6 A5 A4 A3 A2 A1 A0
Register File A
B15 B14 B13 B12 B11 B10 B9 B8 B7 B6 B5 B4 B3 B2 B1 B0
Register File B
32
Data 1
32
32
Data 2
32
Data Memory Controller
16
16
16
16
0 1 2 3 4 5 6 7
8 9 • • •
Data Address 1
Data Address 2
Internal Data Memory (Byte-Addressable)

NOTE A: See CPU Reference Guide (literature number SPRU189) for more details.

Figure 1. VelocITI ’C6200 CPU

PRODUCT PREVIEW

CPU registers

The ’C6201 CPU has a register-based load/store architecture featuring two general-purpose register files (A and B). All data manipulation instructions operate on data residing in registers. The data is moved between the registers and memory using separate load and store instructions. The general-purpose register files are 32 bits wide, each containing 16 registers. Side A registers have multiple paths to the four side A functional units, and side B registers have multiple data paths to side B functional units. There is a single cross-path between the opposite register files/functional units that allows sharing of data, but for optimal data throughput, side A functional units should operate on data located in side A registers and side B units on side B data. All general-purpose registers can be used with instructions that operate on 40-bit data by grouping the registers in pairs. While most registers can be used with most instructions, some also have specific uses as shown in Figure 2.

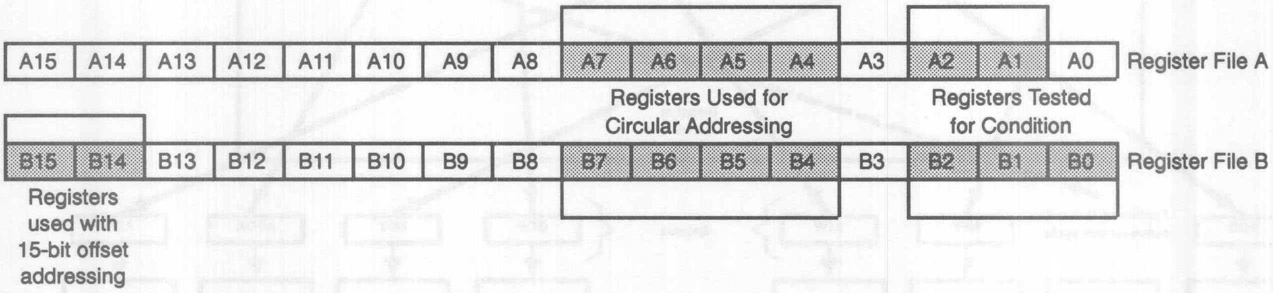

A15
A14
A13
A12
A11
A10
A9
A8
A7
A6
A5
A4
A3
A2
A1
A0
Register File A
Registers Used for Circular Addressing
Registers Tested for Condition
B15
B14
B13
B12
B11
B10
B9
B8
B7
B6
B5
B4
B3
B2
B1
B0
Register File B
Registers used with 15-bit offset addressing

Figure 2. Control Register Usage

In addition to general-purpose register files A and B, the ’C62xx CPU also features a set of control registers used for the following:

- Interrupt control
- Program/cache operation
- Power-down modes
- Data saturation
- Linear/circular and endian-addressing mode setting
- Program counter access
- General-purpose I/O
- Program memory writing

Table 1 lists the functional description for each ’C6201 control register.

CPU registers (continued)

Table 1. Control Registers

ABBREVIATION	NAME	DESCRIPTION
AMR	Addressing-mode register	Specifies whether to use linear or circular addressing for one of eight registers. Also contains block sizes for circular addressing
CSR	Control status register	Contains the global-interrupt-enable bit, cache-control bits, and other miscellaneous control and status bits
PCE1	Program counter	Contains the address of the fetch packet that contains the execute packet in the E1 pipeline stage
IN	General-purpose input register	Contains 32 input signals
OUT	General-purpose output register	Contains 32 output signals
PDATA_O	Program data out	Contains 32 output signals. Used by the STP instruction to write to program space
IER	Interrupt-enable register	Allows enabling/disabling of individual interrupts
IFR	Interrupt-flag register	Displays status of interrupts
ISR	Interrupt-set register	Allows to set pending interrupts manually
ICR	Interrupt-clear register	Allows to clear pending interrupts manually
ISTP	Interrupt-service table pointer	Points to the beginning of the Interrupt-service routine
IRP	Interrupt-return pointer	Contains the address used to return from a maskable interrupt
NRP	Non-maskable-interrupt return pointer	Contains the address used to return from non-maskable interrupt

Figure 3 shows the bit mapping for each of the ’C6201 control registers. Each control register bit map has a corresponding entry in Table 1.

CPU registers (continued)

	31–26	25–21	20–16	15–14	13–12	11–10	9–8	7–6	5–4	3–2	1–0
AMR	rsv	BK1	BK0	B7m	B6m	B5m	B4m	A7m	A6m	A5m	A4m
	Reserved	Block Size 1	Block Size 0	Linear- or Circular-Addressing Mode Select							

	31–24	23–16	15–10	9	8	7–5	4–2	1	0
CSR	CPU ID	REVISION ID	PWRD	SAT	EN	PCC	DCC†	PGIE	GIE
	CPU Type Identification	Silicon identification	Power-Down Modes	Saturate Status	Endian Control	Program/Cache Control	Data/Cache Control	Previous GIE State	Global-Interrupt Enable

† Not used on TMX320C6201 (reserved)

	31–0
PCE1	Program counter (address of fetch packet containing execute packet in the E1 pipeline stage)
IN	General-purpose input (reflects status of 32 general-purpose input pins)
OUT	General-purpose output (drives 32 general-purpose output pins)
PDATA_O	Program data out (contains 32 bits used by the STP instruction to write to program space)

	31–16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
IER	Reserved	IE15	IE14	IE13	IE12	IE11	IE10	IE9	IE8	IE7	IE6	IE5	IE4	rsv	rsv	NMIE	0

	31–16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
IFR	Reserved	IF15	IF14	IF13	IF12	IF11	IF10	IF9	IF8	IF7	IF6	IF5	IF4	rsv	rsv	NMIF	0

	31–16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ISR	Reserved	IS15	IS14	IS13	IS12	IS11	IS10	IS9	IS8	IS7	IS6	IS5	IS4	rsv	rsv	rsv	rsv

	31–16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ICR	Reserved	IC15	IC14	IC13	IC12	IC11	IC10	IC9	IC8	IC7	IC6	IC5	IC4	rsv	rsv	rsv	rsv

	31–10	9–5	4	3	2	1	0
ISTP	ISTB	HPENT	0	0	0	0	0
	Interrupt-Service Table Base	Highest Priority IE-Enabled Interrupt	Interrupt-Service Packet Size				
	Address of the interrupt service fetch packet (ISFP) for the highest priority pending interrupt						

	31–0
IRP	Interrupt Return Address

	31–0
NRP	Non-maskable Interrupt Return Address

NOTE A: Only 0s can be written to reserved fields of the registers.

Figure 3. Control Register Usage

CPU registers (continued)

Table 2 lists the instructions associated with the CPU functional units.

Table 2. Instruction Set Summary

FUNCTION UNITS						
.L UNIT		**.M UNIT**	**.S UNIT**			**.D UNIT**
ABS	NORM	MPY	ADD	EXT	SET	ADD
ADD	NOT	SMPY	ADDK	EXTU	SHL	ADDA
AND	OR		ADD2	MVC †	SHR	LD mem
CMPEQ	SADD		AND	MV	SHRU	LD mem (15-bit offset) ‡
CMPGT	SAT		B disp	MVK	SSHL	MV
CMPGTU	SSUB		B IRP †	MVKH	STP †	NEG
CMPLT	SUB		B NRP†	NEG	SUB	ST mem
CMPLTU	SUBC		B reg	NOT	SUB2	ST mem (15-bit offset) ‡
LMBD	XOR		CLR	OR	XOR	SUB
MV	ZERO				ZERO	SUBA
NEG						ZERO

† .S2 only
‡ .D2 only

Table 3 lists each pipeline phase and the phase descriptions. The pipeline-stage symbols shown can be used to interpret the instruction-execute phases shown in Table 4.

Table 3. Pipeline Description

PIPELINE PHASE	PIPELINE STAGE	SYMBOL	DURING THIS PHASE:
Program Fetch	Program Address Generate	PG	Address of the fetch packet is determined.
	Program Address Send	PS	Address of the fetch packet is sent to memory.
	Program Wait	PW	Program memory read is performed.
	Program Data Receive	PR	Fetch packet is expected at CPU boundary.
Program Decode	Execute Packet Dispatch	DP	Next execute packet is sent to functional units.
	Decode	DC	Instructions are decoded in functional units.
Execute	Execute 1	E1	Instruction conditions are evaluated, operands read. Load/store addresses are computed/modified. Branches affect fetch packet in PG stage. Single-cycle results are written to register file.
	Execute 2	E2	Load address is sent to memory. Store/STP address and data are sent to memory. Single-cycyle instructions can set the SAT bit. Multiply results are written to the register file.
	Execute 3	E3	Load memory reads continue. Multi-cycle instruction can set the SAT bit.
	Execute 4	E4	Load data arrives at the CPU boundary.
	Execute 5	E5	Load data is placed in the register.

Table 4 shows the delay slots and execute stages used by the ’C6201.

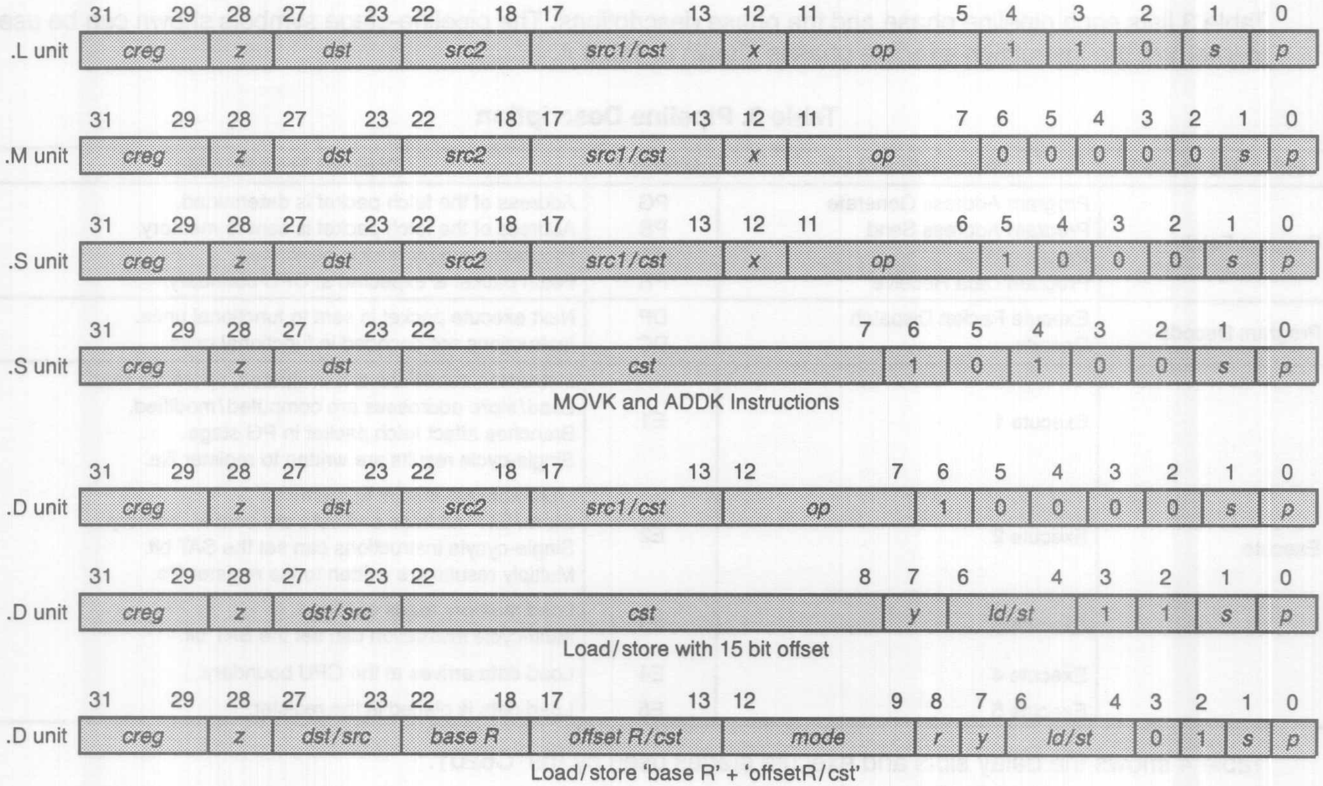
CPU registers (continued)

Table 4. Instruction-Execute Phases

INSTRUCTION TYPE		NUMBER OF DELAY SLOTS	EXECUTE STAGES USED
Branch	(from E1 to E1)	5	E1, , , , , E1
Load	(incoming data)	4	E1, E2, E3, E4, E5
Multiply		1	E1, E2
Load	(address modification)	0	E1
Single-cycle	(ADD, for example)	0	E1
Store		0	E1
STP		0	E1
NOP	(no execution)	—	—

’C6200 CPU instruction opcode map

Figure 4 shows the instruction operation code (opcode) mappings used by the C62xx functional units.


.L unit: 31 29 28 27 23 22 18 17 13 12 11 5 4 3 2 1 0
creg | z | dst | src2 | src1/cst | x | op | 1 | 1 | 0 | s | p
.M unit: 31 29 28 27 23 22 18 17 13 12 11 7 6 5 4 3 2 1 0
creg | z | dst | src2 | src1/cst | x | op | 0 | 0 | 0 | 0 | 0 | s | p
.S unit: 31 29 28 27 23 22 18 17 13 12 11 6 5 4 3 2 1 0
creg | z | dst | src2 | src1/cst | x | op | 1 | 0 | 0 | 0 | s | p
.S unit: 31 29 28 27 23 22 7 6 5 4 3 2 1 0
creg | z | dst | cst | 1 | 0 | 1 | 0 | 0 | s | p
MOVK and ADDK Instructions
.D unit: 31 29 28 27 23 22 18 17 13 12 7 6 5 4 3 2 1 0
creg | z | dst | src2 | src1/cst | op | 1 | 0 | 0 | 0 | 0 | s | p
.D unit: 31 29 28 27 23 22 8 7 6 4 3 2 1 0
creg | z | dst/src | cst | y | ld/st | 1 | 1 | s | p
Load/store with 15 bit offset
.D unit: 31 29 28 27 23 22 18 17 13 12 9 8 7 6 4 3 2 1 0
creg | z | dst/src | base R | offset R/cst | mode | r | y | ld/st | 0 | 1 | s | p
Load/store ‘base R’ + ‘offsetR/cst’

Legend:

p	Parallel instruction	creg	Condition register (A1, A2, B1, B2, B3, 0000b – always true)
s	Destination file A or B	dst	Destination register
x	Cross path	op	Functional unit operation code
y	Base addr in B14 or B15	cst	Constant
z	Test for zero or non-zero	ld/st	Types of load/store (signed, unsigned, half)
r	Reserved	mode	Addressing mode (linear, circular)
src1/src2	Source registers		

Figure 4. Instruction Operation Code Map

memory map

Figure 5 shows the memory map of the TMX320C6201 DSP. The total memory-address range is 4G bytes (corresponding to 32-bit internal address representation). The memory map is divided between the internal-program memory, internal-data memory, three external-memory spaces, and internal-peripheral space. A section of internal memory can be accessed by the host-port interface (HPI).

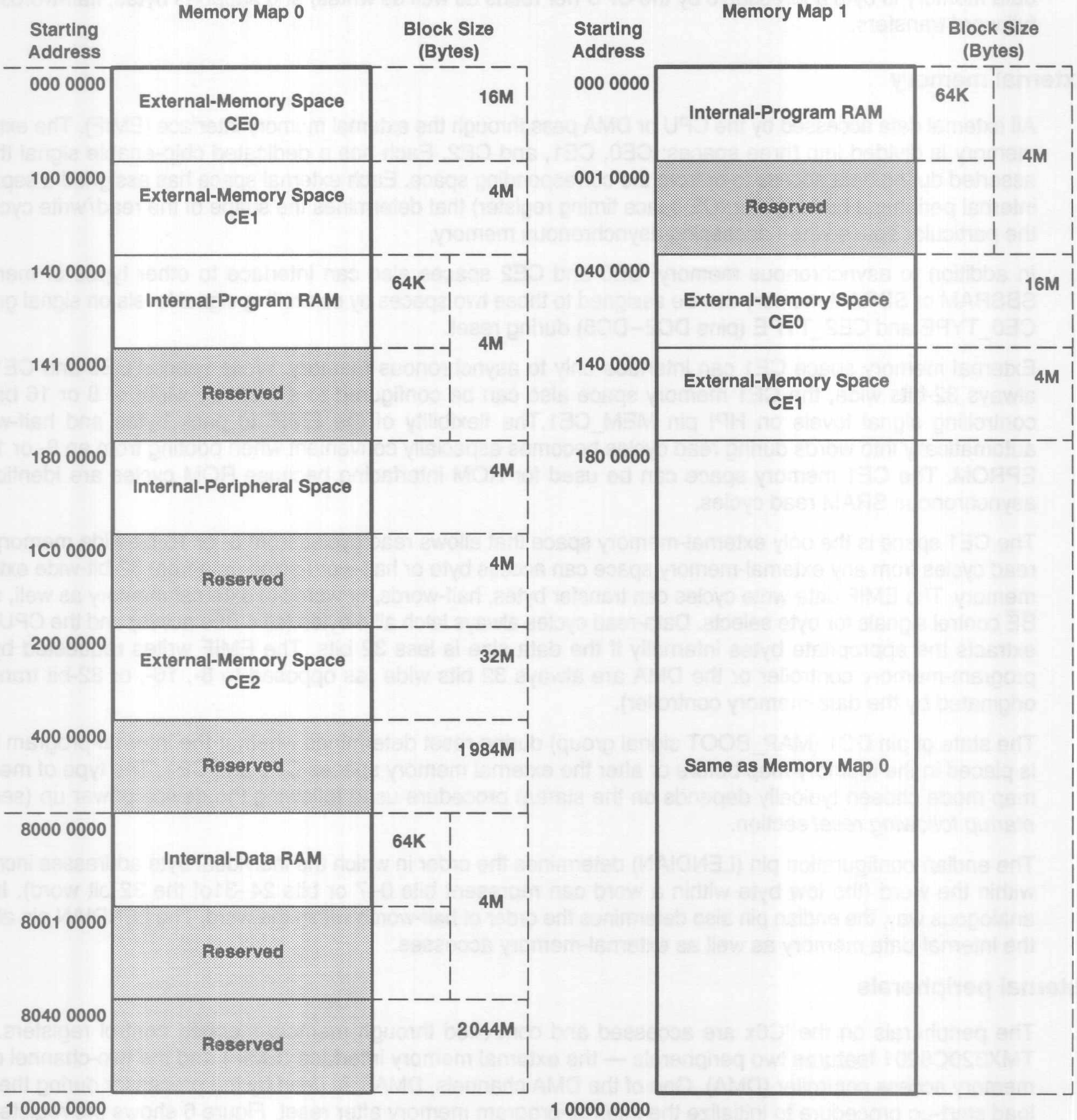


Figure 5. TMX320C6201 Memory Map

internal memory

The internal memory consists of 512K bits of on-chip program/cache memory and 512K bits of on-chip data memory. The program memory, configurable as cache or program, is organized in 2K of 256-bit fetch packets. The ’C6201 fetches all instructions one fetch packet at a time. The packets are processed at the maximum rate of one packet (eight 32-bit instructions) per CPU cycle or at a minimum of one instruction per cycle. The internal data memory is byte addressable by the CPU (for reads as well as writes) and supports bytes, half-words, and full word transfers.

external memory

All external data accessed by the CPU or DMA pass through the external memory interface (EMIF). The external memory is divided into three spaces: CE0, CE1, and CE2. Each has a dedicated chip-enable signal that is asserted during data access to or from the corresponding space. Each external space has assigned a separate internal peripheral bus register (CE space timing register) that determines the shape of the read/write cycle for the particular space when accessing asynchronous memory.

In addition to asynchronous memory, CE0 and CE2 spaces also can interface to other types of memory. SBSRAM or SDRAM memory can be assigned to those two spaces by controlling signal levels on signal groups CE0_TYPE and CE2_TYPE (pins DC2–DC5) during reset.

External memory space CE1 can interface only to asynchronous memory. While spaces CE0 and CE1 are always 32-bits wide, the CE1 memory space also can be configured to a memory width of 8 or 16 bits by controlling signal levels on HPI pin MEM_CE1.The flexibility of the EMIF to pack bytes and half-words automatically into words during read cycles becomes especially convenient when booting from an 8- or 16-bit EPROM. The CE1 memory space can be used for ROM interfacing because ROM cycles are identical to asynchronous SRAM read cycles.

The CE1 space is the only external-memory space that allows read cycles from 8- or 16-bit-wide memory, but read cycles from any external-memory space can access byte or half-word-sized data from 32-bit-wide external memory. The EMIF data-write cycles can transfer bytes, half-words, or words to external memory as well, using $\overline{\text{BE}}$ control signals for byte selects. Data-read cycles always latch all 4 bytes (all 4 $\overline{\text{BE}}$s active) and the CPU then extracts the appropriate bytes internally if the data size is less 32 bits. The EMIF writes requested by the program-memory controller or the DMA are always 32 bits wide (as opposed to 8-, 16-, or 32-bit transfers originated by the data-memory controller).

The state of pin DC1 (MAP_BOOT signal group) during reset determines whether the internal-program RAM is placed in the memory map before or after the external memory spaces CE0 and CE1. The type of memory map mode chosen typically depends on the startup procedure used following the device power up (see the *startup following reset* section.

The endian-configuration pin (LENDIAN) determines the order in which the individual byte addresses increase within the word (the low byte within a word can represent bits 0–7 or bits 24–31of the 32-bit word). In the analogous way, the endian pin also determines the order of half-words within the word. The LENDIAN pin affects the internal-data memory as well as external-memory accesses.

internal peripherals

The peripherals on the ’C6x are accessed and controlled through memory-mapped control registers. The TMX320C6201 features two peripherals — the external memory interface (EMIF) and the two-channel direct memory access controller (DMA). One of the DMA channels, DMA0, is used by the processor during the boot load start-up procedure to initialize the internal-program memory after reset. Figure 6 shows the registers.

internal peripherals (continued)

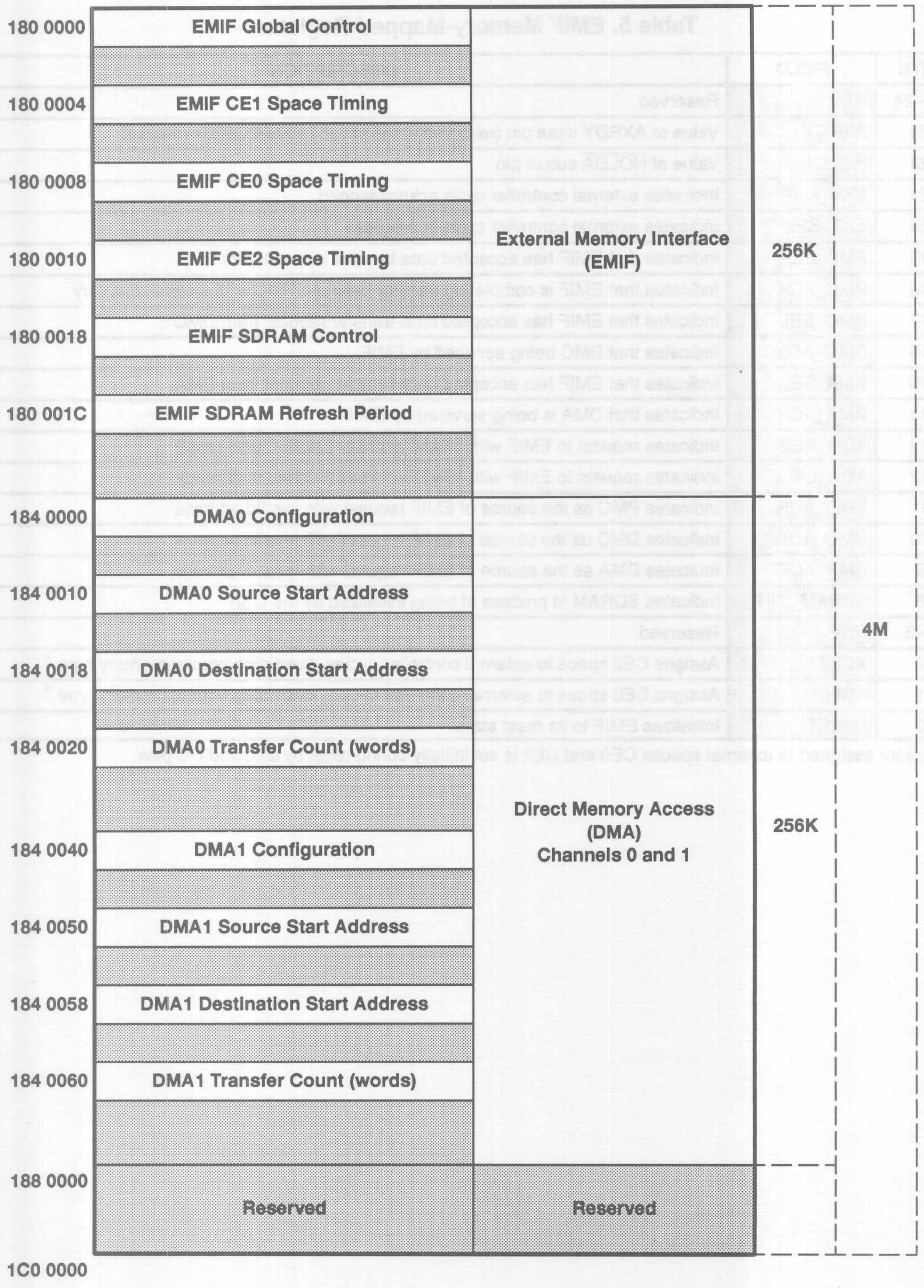


Figure 6. Internal Peripheral Space Memory-Mapped Registers

memory-mapped registers – EMIF

Table 5. EMIF Memory-Mapped Registers

	BITS	FIELD	DESCRIPTION
EMIF Global Control Register	31–24	RSV	Reserved
	23	AXRDY	Value of AXRDY input pin (reserved in the initial TMX320C6201 release)
	22	HOLDA	Value of $\overline{\text{HOLDA}}$ output pin
	21	EXT_ACK	Indicates external controller cycle acknowledged
	20	EXT_SEL	Indicates external controller cycle in progress
	19	PMC_SEL	Indicates that EMIF has accepted data transfer request from PMC
	18	PMC_ACK	Indicates that EMIF is completing transfer between PMC and external memory
	17	DMC_SEL	Indicates that EMIF has accepted data-transfer request from DMC
	16	DMC_ACK	Indicates that DMC being serviced by EMIF
	15	DMA_SEL	Indicates that EMIF has accepted data-transfer request from DMA
	14	DMA_ACK	Indicates that DMA is being serviced by EMIF
	13	ADR_RES	Indicates request to EMIF with invalid address (latched until reset)
	12	ADR_LAT	Indicates request to EMIF with invalid address (latched until read)
	11	PMC_ADR	Indicates PMC as the source of EMIF request with invalid address
	10	DMC_ADR	Indicates DMC as the source of EMIF request with invalid address
	9	DMA_ADR	Indicates DMA as the source of EMIF request with invalid address
	8	SDRAM_INIT	Indicates SDRAM in process of being initialized by the DSP
	7–3	RSV	Reserved
	2	XCE2	Assigns CE2 space to external controller cycles, overriding normal memory type †
	1	XCE0	Assigns CE0 space to external controller cycles, overriding normal memory type †
	0	RESET	Initializes EMIF to its reset state

† The type of memory assigned to external spaces CE0 and CE2 is set initially during reset by selected DC pins.

PRODUCT PREVIEW

memory-mapped registers – EMIF(continued)

CE Space Timing Register	31–20	19–16	15–14	13–8	7–2	1–0
	RSV†	SETUP‡	RSV	STROBE‡	RSV	HOLD‡
	Reserved	Setup Time Before Strobe Falling Edge	Reserved	Active Strobe Width	Reserved	Hold Time After Strobe Rising Edge

SDRAM Control Register	31–25	24	23–20	19–16	15–12	11–10	9	8–7	6–4	3	2–0
			Used by the 320C6201			Used by the SDRAM					
	RSV	INIT§	TRCD	TRP	TRC	RSV	WBL†	RSV	RL	BT†	BL
	Reserved	Initialize SDRAM Internal Mode Register	SDRAM T_{RCD} Parameter	SDRAM T_{RP} Parameter	SDRAM T_{RC} Parameter	Reserved	Write Burst Length (RSV)	Reserved	Read Latency	Burst Type (RSV)	Burst Length
			In CLKOUT2 Cycles								

SDRAM Refresh Period Register	31–12	11–4	3–0
	RSV	PERIOD_1	PERIOD_0
	Reserved	Timer High-Counter Reload Value	Timer Low-Counter Reload Value

† Only 0s can be written to reserved fields of the registers.
‡ The smallest value permitted for this field is 1.
§ Bits 0–11 of the SDRAM control register are used to initialize the SDRAM internal mode register.

Figure 7. EMIF Memory-Mapped Registers

PRODUCT PREVIEW

memory mapped registers – DMA

	31 26	25	24	23 19	18 14	13 8	7 6	5 4	3 2	1 0
	RSV	TCINIT	RSV	WSYNC	RSYNC	RSV	DST_DIR	SRC_DIR	STATUS	START
DMA Control Register	(see Note A)	Enables interrupt generation following transfer		Identifies interrupt synchronizing the writes (or no sync)	Identifies interrupt synchronizing the reads (or no sync)		Increment or decrement destination address after each transfer	Increment or decrement source address after each transfer	Shows if DMA is started or stopped	Starts, pauses, aborts DMA operation

	31	30 25	24 2	1 0
DMA Source Address Register	SA	RSV	SRC_ADR	RSV
	Source addr select: internal-data memory or external-data memory /internal-program memory	These bits are 0 to cover EMIF and internal program memory spaces	Source-Word Address	These bits are 0 since DMA supports only word-sized transfers

	31	30 25	24 2	1 0
DMA Destination Address Register	SA	RSV	SRC_ADR	RSV
	Destination addr select: internal data memory or external data memory /internal program memory	These bits are 0 to cover EMIF and internal program memory spaces	Destination Word Address	These bits are 0 since DMA only supports word-sized transfers

	31 16	15 0
DMA-Transfer Counter Register	RSV	TC
	Reserved	DMA Transfer Counter (see Note B)

NOTES: A. Only 0s are permitted to be written to reserved fields of the registers.
B. The transfer counter is decremented upon the completion of the write portion of each transfer.

Figure 8. DMA Memory-Mapped Registers

internal data memory addressing

The 512K bits of internal data memory can be accessed both from the CPU and through the host-port interface (HPI). The CPU sees the internal data memory as 64K bytes, organized in eight columns, each 1 byte wide and 16K rows tall. The HPI addresses the same memory across four columns, each 1 half-word wide and 16K rows tall (32K half-words total). Unlike the CPU, the HPI only has the capability to address the internal data in a little-endian mode where the address of individual half-words grows in the same direction as the addresses of the full words. See Figure 9 for the addressing example.

NOTE:

Bit 16 of the host-port address (pin HA15) is not used in the initial release of the device (it must be driven high or low and not left floating).

Internal Data Memory As Addressed By the CPU (Little Endian)

8000 0000	8000 0001	8000 0002	8000 0003	8000 0004	8000 0005	8000 0006	8000 0007
8000 0008	8000 0009	8000 000A	8000 000B	8000 000C	8000 000D	8000 000E	8000 000F
•	•	•	•	•	•	•	•
•	•	•	•	•	•	•	•
•	•	•	•	•	•	•	•
8000 FFF0	8000 FFF1	8000 FFF2	8000 FFF3	8000 FFF4	8000 FFF5	8000 FFF6	8000 FFF7
8000 FFF8	8000 FFF9	8000 FFFA	8000 FFFB	8000 FFFC	8000 FFFD	8000 FFFE	8000 FFFF

64K bytes

(byte addressable)

Same Internal Data Memory As Addressed Through the HPI

0000	0001	0002	0003
0004	0005	0006	0007
•	•	•	•
•	•	•	•
•	•	•	•
7FF8	7FF9	7FFA	7FFB
7FFC	7FFD	7FFE	7FFF

32K

half-words

(half-word addressable)

Figure 9. Internal Memory Addressing

PRODUCT PREVIEW

simultaneous internal memory access from both sides of the CPU

To visualize maximum memory-access performance, the internal data memory of the 320C6201 can be organized in four blocks, each 2 bytes wide and 16K rows tall. As shown in Figure 10 and Figure 11, the data is byte-addressable by the CPU with address increasing across the blocks from left to right (this example uses little-endian mode). For both sides of the CPU to access data simultaneously during the same CPU cycle, none of the bytes of side A data can reside in the same memory block as any of the side B data bytes. As shown in Figure 10, a word of data is loaded to a side A register at the same time as another word is stored from a side B register. The two accesses in Figure 11 (both 8 bits wide), however, take two cycles because both bytes reside in the same block of the internal data memory.

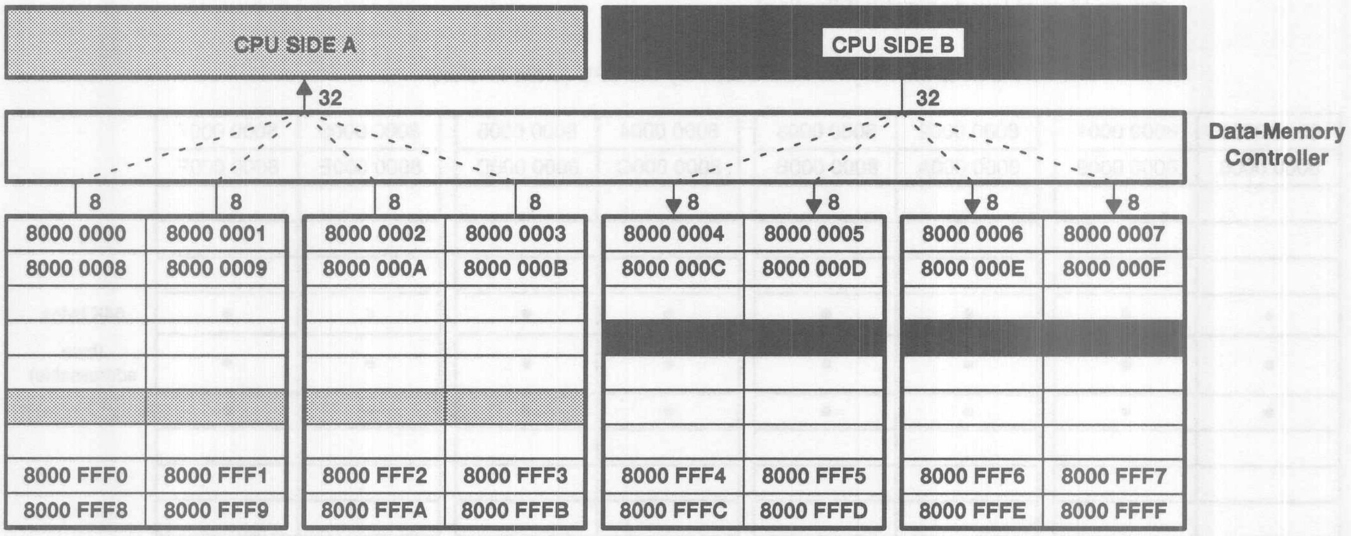


Figure 10. Simultaneous Internal Memory Access

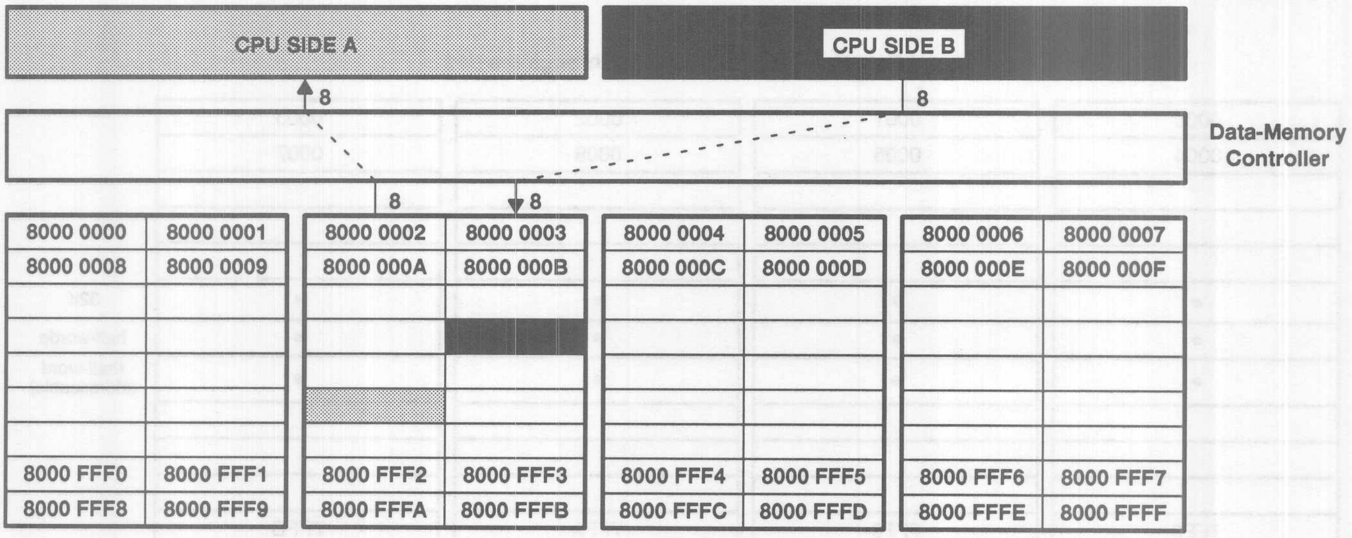


Figure 11. Two-Cycle Internal Memory Access

processor reset

Device reset begins immediately following the falling signal edge at the RESET pin. During reset, all functional blocks of the processor are initialized to a known state as are all CPU and internal peripheral bus registers. The EMIF-output signals (except HOLDA) and all HPI-output signals are put in the high-impedance state immediately following the falling edge of RESET. They remain in the high-impedance state until the RESET signal returns to a high level (see Figure 40). One of the signals, HACK, is driven low for the duration of the reset signal.

Most of the HPI address pins have a special function during device reset. They become EMIF configuration pins as shown on Figure 12. The lowest bit HPI-address pin becomes a memory-map-select pin (it also selects the boot DMA transfer source address in future revisions). The selection of low or high state for that pin typically depends on the processor start-up mode after reset (pins DC 13 to 11).

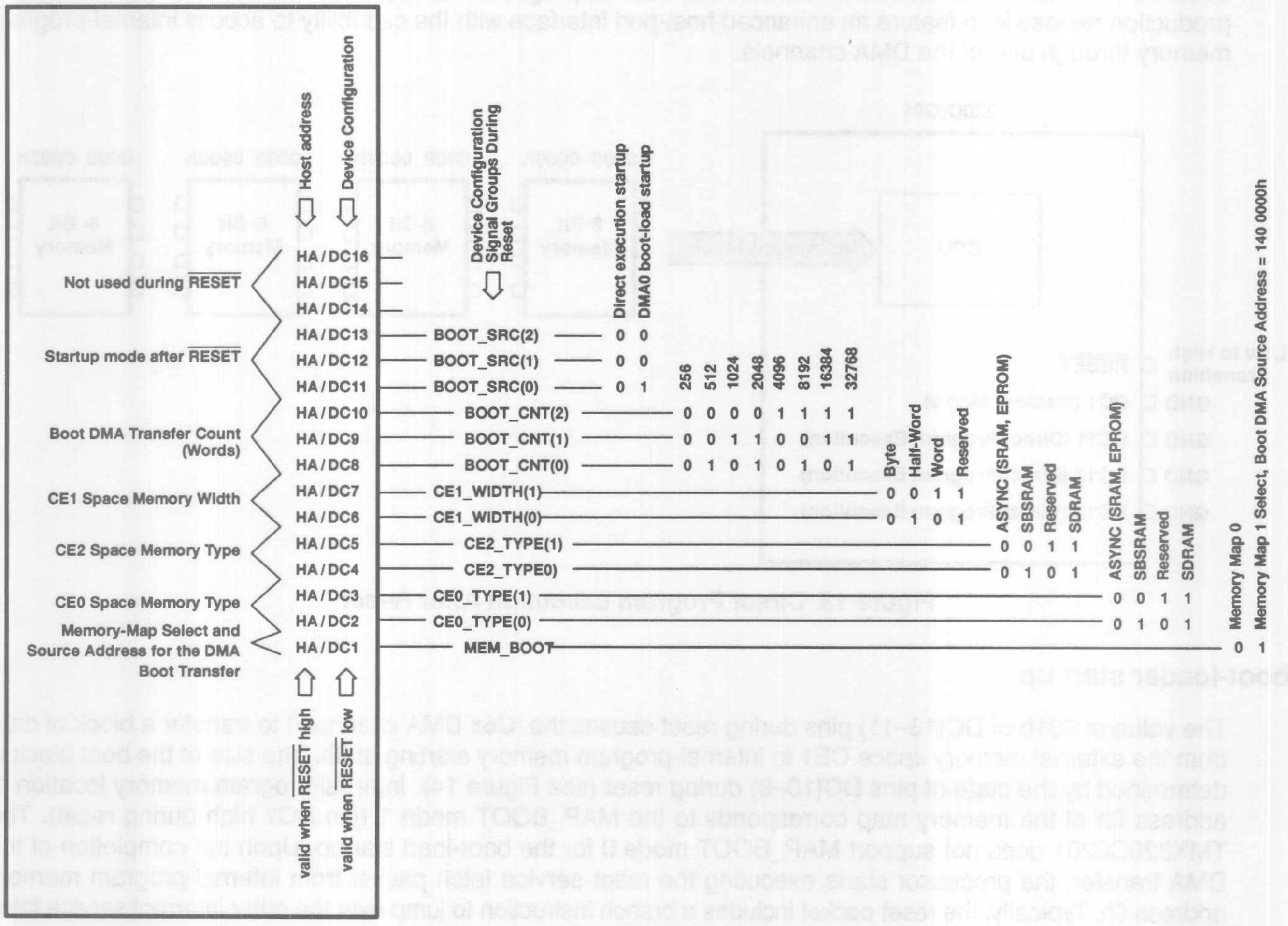


Figure 12. Device-Configuration Pins During Reset

start-up operation after reset

There are two methods by which the 'C6201 can start operating following the deassertion of the $\overline{\text{RESET}}$ signal. The state of signal DC11 (during reset) determines if the device starts direct program execution from external address 0h or whether it first uses an internal DMA channel 0 to boot-load the internal program memory from an external EPROM before commencing program execution from internal address 0h.

direct program execution

The value of 000b of DC(13–11) pins during reset causes the '320C6201 to start program execution directly from memory starting at address 0h (location of the reset-service-fetch packet). Depending on the MAP_BOOT mode used, address 0h falls inside or external to the chip. Typically MAP_BOOT mode 0 (pin DC1 low during reset) is used to start execution from 32-bit external EPROM or SRAM that has been properly initialized by the host during reset. Memory map 1 mode typically is not used with the TMX320C6201 for direct execution startup, because there is no mechanism to initialize the internal program memory other than by the 'C6x itself. The production release is to feature an enhanced host-port interface with the capability to access internal-program memory through one of the DMA channels.

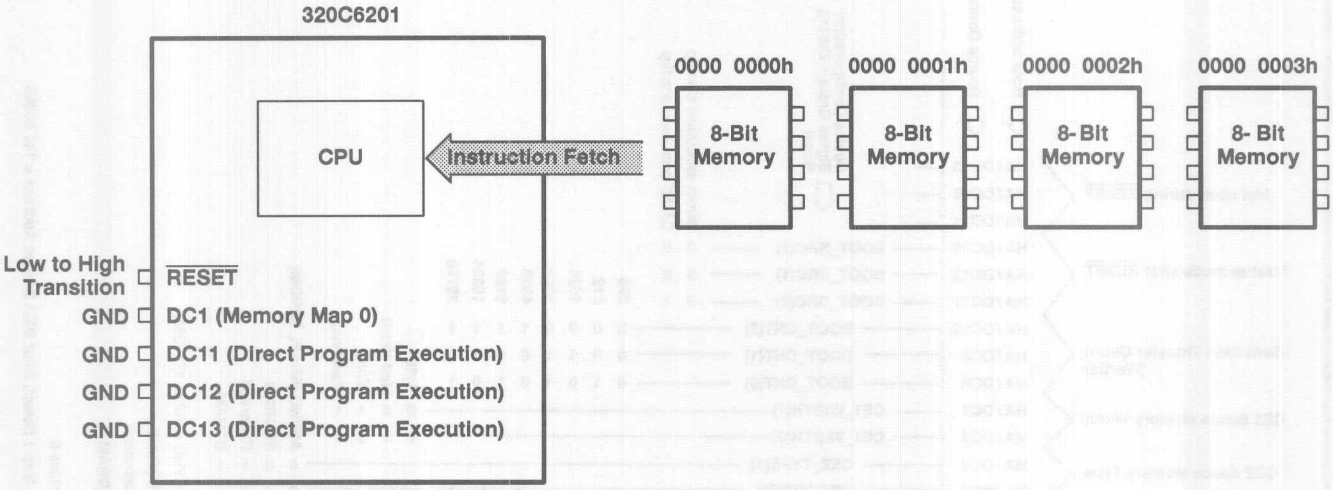


Figure 13. Direct Program Execution After Reset

boot-loader start up

The value of 001b of DC(13–11) pins during reset causes the 'C6x DMA channel 0 to transfer a block of data from the external memory space CE1 to internal-program memory starting at 0h. The size of the boot block is determined by the state of pins DC(10–8) during reset (see Figure 14). Internal-program memory location at address 0h of the memory map corresponds to the MAP_BOOT mode 1 (pin DC1 high during reset). The TMX320C6201 does not support MAP_BOOT mode 0 for the boot-load startup. Upon the completion of the DMA transfer, the processor starts executing the reset service fetch packet from internal-program memory address 0h. Typically, the reset packet includes a branch instruction to jump over the other interrupt service fetch packets (ISFPs) to the application boot-loader program. The boot loader then loads the internal data memory with application-initialized data and loads another program block to some location in the external memory space CE0 or CE2 before branching to the main program loop.

boot-loader start up (continued)

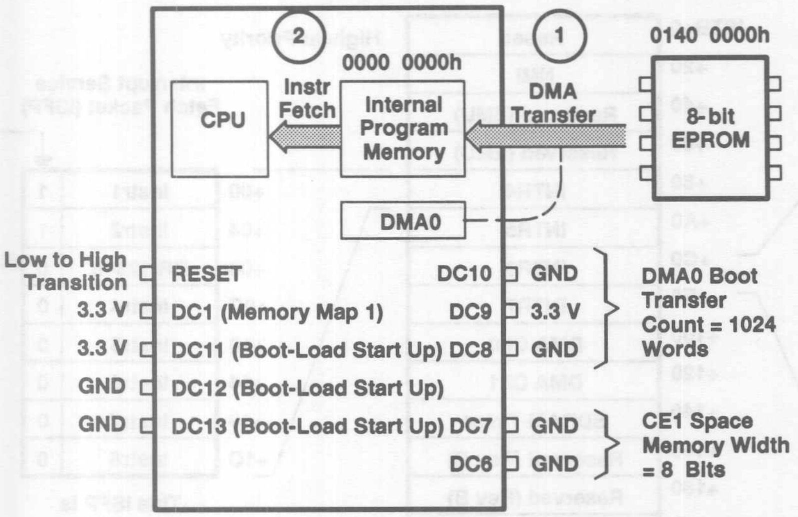


Figure 14. Boot-Load Start-Up Mode After Reset

interrupt servicing

Before an interrupt can be processed automatically, its corresponding bit must be set in the interrupt-enable register. the IER NMIE bit must be set and the global-interrupt-enable bit (GIE) must be set as well in the control status register (CSR). During interrupt processing, the processor stops current program execution and saves the address of the next instruction scheduled for execution in the Interrupt-return pointer (IRP). In the case of the non-maskable interrupt (NMI), the return address is saved in the NRP register. Program execution is then redirected to one of 16 pre-assigned contiguous locations in the interrupt service table (IST) The IST base is assigned to address 0h following reset, but can be relocated anywhere on a 1K-byte address boundary by changing the ISTB field of the ISTP register). The ISFPs consist of eight instructions of which one has to be a branch to the the IRP address (to return from interrupt) or a branch to additional code if the interrupt service routine (ISR) is larger then the ISFP (see Figure 15). The branch must be executed at least five cycles before the end of the ISFP to prevent intrusion into ISFPs belonging to other interrupts. This can be accomplished either by placing the branch five non-parallel instructions before the end of the ISFP or by following the branch with a *NOP 5* instruction.

With the GIE bit disabled, the highest priority active interrupt still can be identified in software by polling the ISTP register containing the address pointing to the ISFP of the next interrupt in line for processing if the GIE were enabled. All active interrupts also can be identified by polling the interrupt flag register (IFR). Interrupt set and clear registers (ISR and ICR) can be used to set/clear interrupts manually in software.

During automatic-interrupt processing, the IFR bit is cleared, and the active interrupt is encoded on the four INUM pins, at the beginning of the IACK pulse (see the *timing section* of the data sheet). At the same time, the GIE bit is copied to the PGIE field of CSR, and the GIE bit is cleared for all interrupts except NMI. The PGIE bit is copied back to GIE field upon return from interrupt through the *B IRP* instruction (at the end of the E1 phase). Returning from NMI interrupt by the way of the *B NRP* instruction does not copy the PGIE bit to the GIE field.

interrupt servicing (continued)

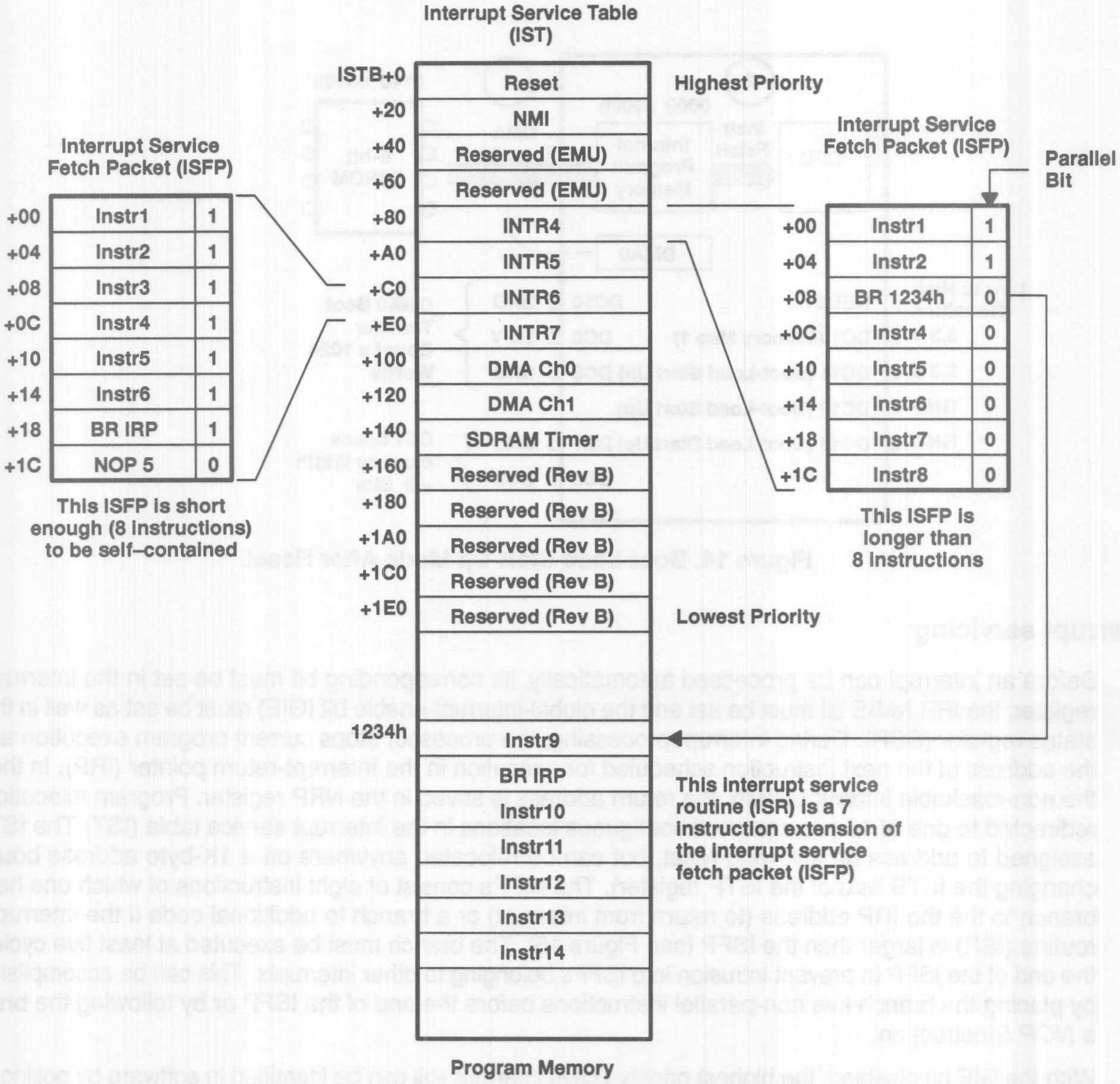


Figure 15. Interrupt Processing

clock characteristics and timing

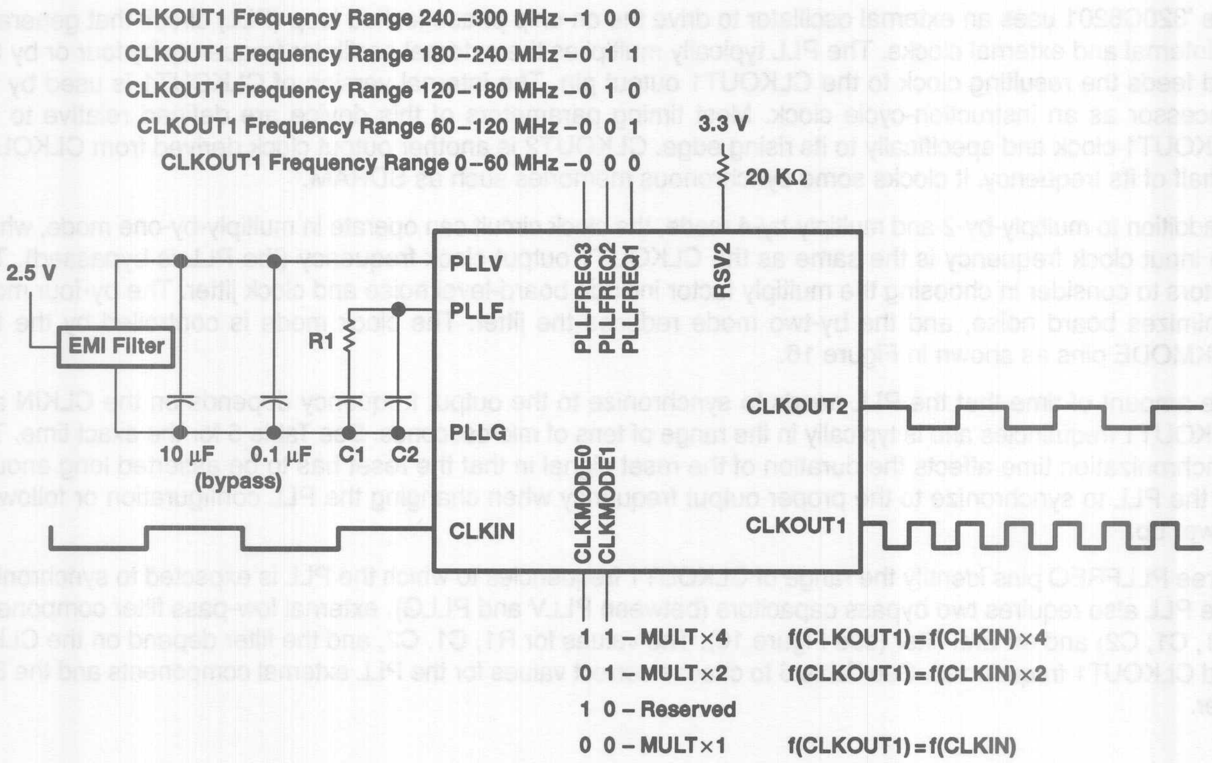
The '320C6201 uses an external oscillator to drive the on-chip phase-locked loop (PLL) circuit that generates all internal and external clocks. The PLL typically multiplies the external oscillator frequency by four or by two and feeds the resulting clock to the CLKOUT1 output pin. The internal version of CLKOUT1 is used by the processor as an instruction-cycle clock. Most timing parameters of this device are defined relative to the CLKOUT1 clock and specifically to its rising edge. CLKOUT2 is another output clock derived from CLKOUT1 at half of its frequency. It clocks some synchronous memories such as SDRAM.

In addition to multiply-by-2 and multiply-by-4 mode, the clock circuit can operate in multiply-by-one mode, where the input clock frequency is the same as the CLKOUT1 output-clock frequency (the PLL is bypassed). The factors to consider in choosing the multiply factor include board-level noise and clock jitter. The by-four mode minimizes board noise, and the by-two mode reduces the jitter. The clock mode is controlled by the two CLKMODE pins as shown in Figure 16.

The amount of time that the PLL needs to synchronize to the output frequency depends on the CLKIN and CLKOUT1 frequencies and is typically in the range of tens of microseconds. See Table 6 for the exact time. The synchronization time affects the duration of the reset signal in that the reset has to be asserted long enough for the PLL to synchronize to the proper output frequency when changing the PLL configuration or following power up.

Three PLLFREQ pins identify the range of CLKOUT1 frequencies to which the PLL is expected to synchronize. The PLL also requires two bypass capacitors (between PLLV and PLLG), external low-pass filter components (R1, C1, C2) and an EMI filter (see Figure 16). The values for R1, C1, C2, and the filter depend on the CLKIN and CLKOUT1 frequencies. See Table 6 to choose correct values for the PLL external components and the EMI filter.

clock characteristics and timing (continued)



NOTES: A. In CLKMODE x2 and x4, values for C1, C2, and R2 depend on CLKIN and CLKOUT1 frequencies.

B. For CLKMODE x1 the PLL circuit is bypassed and the choice of component values is irrelevant and the three analog filter pins (PLLV, PLLF, PLLG) can remain unconnected.

Figure 16. Clock PLL-Functional Diagram

clock characteristics and timing (continued)

Table 6. PLL-Component-Selection Table†

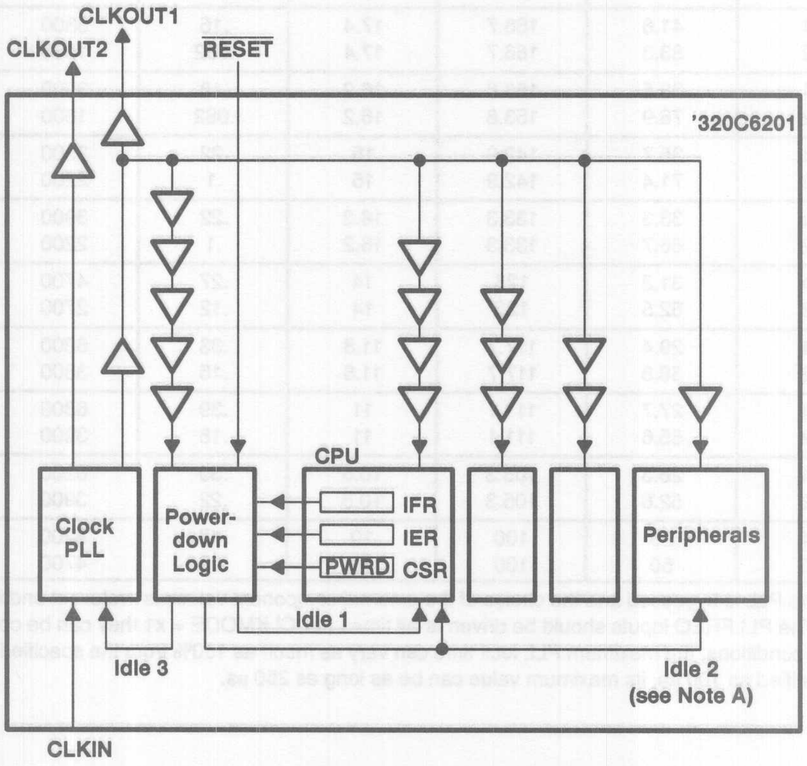
CYCLE TIME (ns)	CLKMODE	CLKIN (MHz)	CLKOUT1 (MHz)	R1 (Ω)	C1 (µF)	C2 (pF)	EMI FILTER (PART NO.)	LOCK TIME‡ (µs)
5	x4	50	200	16.9	.15	2700	TDK #153	59
5	x2	100	200	16.9	.068	1500	TDK #153	28
5.5	x4	45.5	181.8	13.7	.18	3900	TDK #153	49
5.5	x2	90.9	181.8	13.7	0.82	1800	TDK #153	23
6	x4	41.6	166.7	17.4	.15	3300	TDK #153	68
6	x2	83.3	166.7	17.4	.082	1500	TDK #153	36
6.5	x4	38.5	153.8	16.2	.18	3900	TDK #153	70
6.5	x2	76.9	153.8	16.2	.082	1800	TDK #153	33
7	x4	35.7	142.9	15	.22	3900	TDK #153	72
7	x2	71.4	142.9	15	.1	2200	TDK #153	34
7.5	x4	33.3	133.3	16.2	.22	3900	TDK #153	84
7.5	x2	66.7	133.3	16.2	.1	2200	TDK #153	40
8	x4	31.3	125	14	.27	4700	TDK #153	77
8	x2	62.5	125	14	.12	2700	TDK #153	36
8.5	x4	29.4	117.7	11.8	.33	6800	TDK #153	67
8.5	x2	58.8	117.7	11.8	.15	3300	TDK #153	32
9	x4	27.7	111.1	11	.39	6800	TDK #153	68
9	x2	55.6	111.1	11	.18	3900	TDK #153	33
9.5	x4	26.3	105.3	10.5	.39	8200	TDK #153	65
9.5	x2	52.6	105.3	10.5	.22	3900	TDK #153	35
10	x4	25	100	10	.47	8200	TDK #153	68
10	x2	50	100	10	.22	4700	TDK #153	33

† For CLKMODE = x1, the PLL is bypassed and the choice of the external component values is irrelevant and the PLLV, PLLF, and PLLG inputs can be unconnected. The PLLFREQ inputs should be driven at all times (for CLKMODE = x1 they can be connected to GND).

‡ Under some operating conditions, the maximum PLL lock time can vary as much as 150% from the specified typical value. For example, if the typical lock time is specified as 100 µs, its maximum value can be as long as 250 µs.

power-down modes (idle1, 2, and 3)

Most of the operating power of CMOS logic is dissipated during circuit switching from one logic state to another. By preventing some or all of chip logic from switching, significant power savings can be realized without losing any data or operational context. Power-down modes idle1 and idle2 block the internal clock inputs at the boundary of the selected blocks of logic, effectively preventing them from switching. Idle 1 shuts down the CPU and Idle 2 shuts down both the CPU and the on-chip peripherals.The idle 2 mode is not available on the initial release (TMX320C6201). Additional power savings are accomplished in power-down mode idle3, where the entire on-chip clock tree (including multiple buffers) is "disconnected" at the clock source that is the clock PLL (see Figure 17).



NOTE A: Power-down mode idle2 is not implemented in the initial release of the '320C6201.

Figure 17. Power-Down Mode Logic

31 ••• 16	15	14	13	12	11	10	9 ••• 0	
▨	Reserved	Enabled or Non-Enabled Interrupt Wake	Enabled Interrupt Wake	Idle3	Idle2	Idle1	▨	CSR
	5	4	3	2	1	0		

PWRD Field of CSR Register

Figure 18. CSR Register Power-Down Modes

power-down modes (idle1, 2 and 3) (continued)

The power-down modes are triggered by setting bits 0–2 in the PWRD field of the control status register (CSR). Idle3 mode can be aborted only by device reset, while idle1 and idle2 modes also can be terminated by an enabled interrupt or any interrupt (enabled or not), as directed by bits 3 and 4 of the PWRD field. When writing to CSR, all bits of the PWRD field should be set at the same time. When more then one logic 1 is written to the PRWD field bits 0–2, the highest order enabled bit determines the power-down mode.

Table 7. Characteristics of the Power-Down Modes (Idle1, 2, and 3)

POWER-DOWN MODE	TRIGGER ACTION	WAKE–UP METHOD[†]	EFFECT ON CHIP'S OPERATION
Idle1	Write a logic 1 into bit 0 of PWRD field of CSR register	Internal interrupt, external interrupt, or reset	CPU halted (except for the interrupt logic)
Idle2	Write a logic 1 into bit 1 of PWRD field of CSR register	External interrupt or reset	Peripherals halted and CPU halted (except for the interrupt logic)
Idle3	Write a logic 1 into bit 2 of PWRD field of CSR register	Reset only	PLL operation is halted, stopping the clock tree from switching and resulting in the entire chip being halted

[†] The wake-up from idle2 or idle3 can be triggered by either an enabled interrupt, or any interrupt (enabled or not). The first case is selected by writing a logic 1 into bit 3 of the CSR PWRD field, and the second case is selected by writing a logic 1 into bit 4 of the same field. When both bits 3 and 4 are set to 1, any interrupt (enabled or not) wakes the processor from idle1 or idle2. If power-down mode is terminated by a non-enabled interrupt, the program execution returns to the instruction following the one that caused the powerdown by setting the idle bits in the CSR. Wake-up by an enabled interrupt executes the corresponding interrupt service routine first, prior to returning to the code aborted by the power down.

thermal performance

Like most high-performance processors, the '320C6201 can dissipate some thermal energy during normal operation. The high level of integration, high clock frequency, and large on-chip memory arrays have an increased effect on the silicon-junction temperature (T_J) compared with a lower performance DSP. To ensure proper operation and device reliability, the silicon-junction temperature must not exceed the maximum temperature (related to maximum case temperature T_{CMAX}). Under operating conditions, system-level thermal management is needed to assist with dissipating the heat away from the chip package. A passive heat sink provides a reliable and cost-effective method for removing the excess heat. The size and shape of the heat sink depend on the operating conditions of the '320C6201 device and the characteristics of the system within which it operates.

- '320C6201 device operating conditions that can affect power dissipation:
  - Operating frequency
  - Chip manufacturing process type
  - Use of power-down (idle) modes
  - Amount of on-chip activity:
    - Number of functional units exercised every cycle
    - Frequency of internal data memory accesses
    - Frequency of internal program memory accesses
    - Rate at which the internal buses are charged and discharged
    - Activity level of on-chip peripherals (DMAs, serial ports, and others)
    - Rate at which data is driven off and on the chip
- System considerations for choosing a heat sink:
  - Air-flow rate over the board housing the '320C6201
  - Ambient-air temperature
  - Area/type of joint between package and heat sink
  - Board design and layout

heat-sink selection procedure

Most of the current consumed by CMOS devices is alternating current (AC), which is charging and discharging the capacitances of internal nodes, pins, and the external pin loads. The current flowing through the '320C6201 causes the temperature of the silicon die to rise. To maintain device reliability and proper operation, the junction temperature must not exceed the maximum specified junction temperature. As the heat is transferred from the die to the package, the case temperature of the package also rises. The junction-to-case thermal resistance RΘjc, together with the maximum junction temperature, are parameters that determine maximum case temperature T_{CMAX}. Because of these factors, the specification of temperature is given in terms of T_{CMAX} in the *recommended operating conditions* section of this document.

From the case, the heat typically flows to the surrounding air up through the heat sink and down through the pins and the board (see Figure 19). The heat dissipation through the board material largely depends on the number of power and ground layers inside the board. The portion of the power that is dissipated by the heat sink must first propagate through the case-sink joint. The thermal resistance of case to sink joint can vary, depending on the type of joint used. The rate at which the heat is transferred from the sink to ambient air depends on the velocity of the air at the heat sink. The heat flow from the sink rises with the speed of the airflow. The objective of the examples in this section is to present a method for first-order approximation of the size of the heat sink needed to keep the maximum case temperature from exceeding the T_{CMAX} during maximum system operating conditions. Following heat sink installation, the actual case temperature must be measured to verify that if does not exceed the T_{CMAX} value.

Heat-transfer analysis can be a complex task depending on the degree of accuracy required in modeling of system components. A number of computational fluid dynamics and heat-transfer tools is available to provide highly accurate results; however, data must be verified using test methods. This document uses a simplified approach to select a heat sink that matches a particular operating environment. The driving parameter in determining the heat flow is the maximum case temperature which must stay below T_{CMAX} at all times during device operation. The system parameters that have first-order effect on the case temperature are the average device power dissipation, ambient air temperature, and air approach velocity. The choice of the heat sink largely depends on those factors as shown in the following examples. Other factors that may affect the heat dissipation include board design/materials and the case-to-sink attachment method. The relatively long time constant of the case/sink combination (around two minutes) can effectively smooth out any power peaks that occur. To validate proper operation of the heat sink, a small hole must be drilled in the center of the sink for placement of a miniature thermocouple directly on the case (center top surface) for measuring the actual case temperature under system maximum expected operating conditions (see Figure 19).

heat-sink selection procedure (continued)

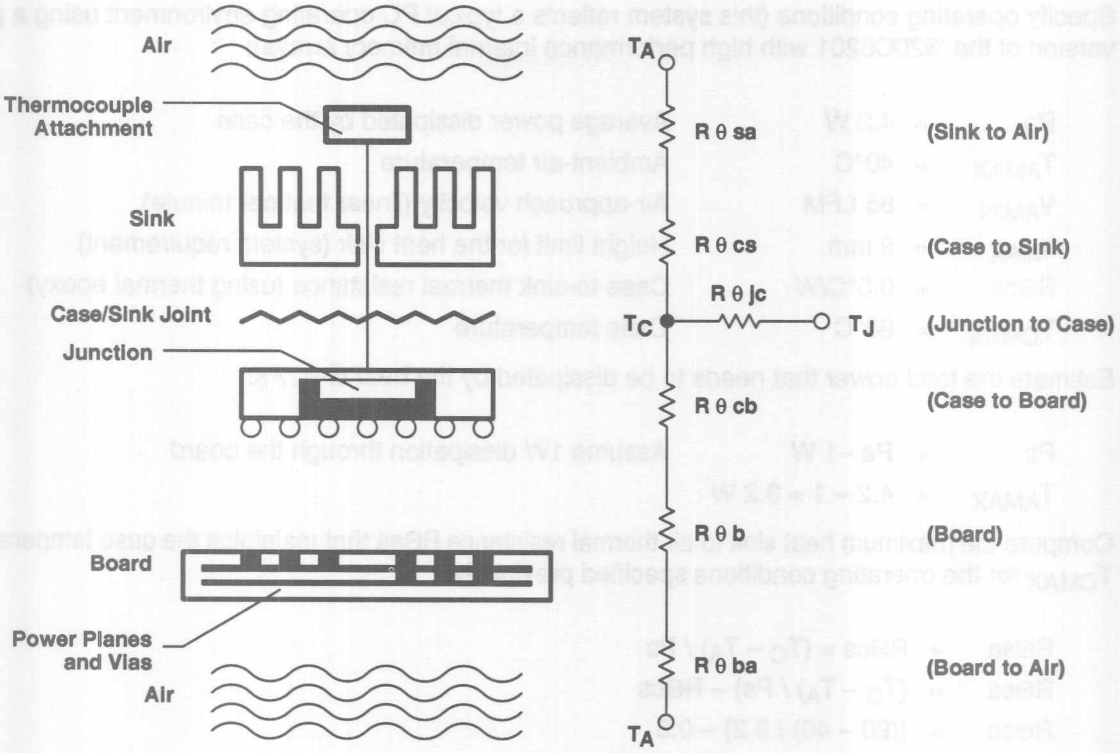


Figure 19. Thermal Model Used for Heat-Flow Analysis

heat sink selection process – example 1

- Specify operating conditions (this system reflects a typical PC operating environment using a production version of the '320C6201 with high performance internal memory arrays):

$P_a = 4.2$ W   Average power dissipated by the case
$T_{AMAX} = 40°C$ Ambient-air temperature
$V_{AMIN} = 85$ LFM   Air-approach velocity (linear feet per minute)
$h_{MAX} = 9$ mm   Height limit for the heat sink (system requirement)
$R\Theta cs = 0.5°C/W$ Case-to-sink thermal resistance (using thermal epoxy)
$T_{CMAX} = 90°C$ Case temperature

- Estimate the total power that needs to be dissipated by the heat sink, Ps:

$Ps = Pa - 1$ W   Assume 1W dissipation through the board
$T_{AMAX} = 4.2 - 1 = 3.2$ W

- Compute the maximum heat sink to air thermal resistance RΘsa that maintains the case temperature below T_{CMAX} for the operating conditions specified previously:

$$R\Theta sa + R\Theta cs = (T_C - T_A) / Ps$$
$$R\Theta sa = \{T_C - T_A) / Ps\} - R\Theta cs$$
$$R\Theta sa = \{(90 - 40) / 3.2\} - 0.5$$
$$R\Theta sa = 15.1°C/W$$

- Choose the heat sink based on the computed maximum sink-to-air resistance RΘsa and the minimum air-approach velocity inside the enclosure. The heat-sink performance plot in Figure 20 verifies that at 85 LFM air velocity, the sink-to-air thermal resistance indeed stays at, or below, the Rsa value of 15.1°C/W that is required to maintain the case temperature below T_{CMAX} for the listed operating conditions. Actual measurement of the case temperature must be performed next to verify heat sink performance.

PRODUCT PREVIEW

heat sink selection process – example 1 (continued)

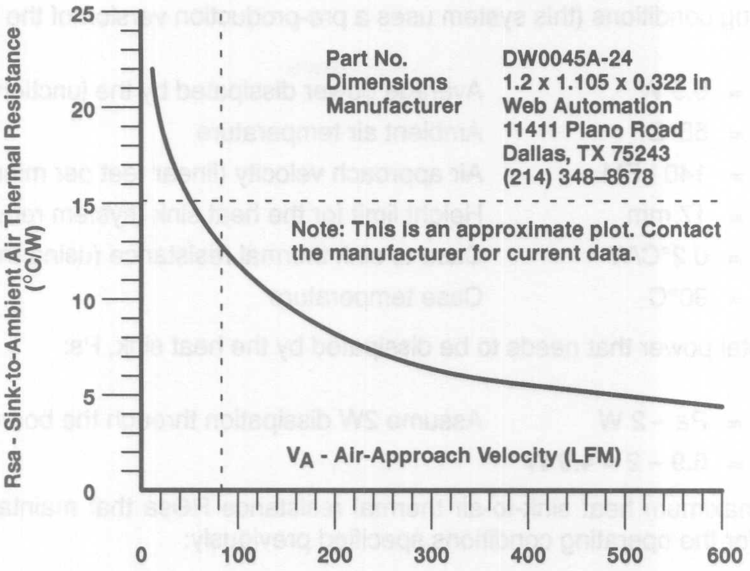


Figure 20. Example 1 Heat-Sink Performance Plot

PRODUCT PREVIEW

heat sink selection process – example 2

- Specify operating conditions (this system uses a pre-production version of the ’320C6201):

Pa	= 6.9 W	Average power dissipated by the junction
T_{AMAX}	= 55°C	Ambient air temperature
V_{AMIN}	= 140 LFM	Air approach velocity (linear feet per minute)
h_{MAX}	= 17 mm	Height limit for the heat sink (system requirement)
RΘcs	= 0.2°C/W	Case to sink thermal resistance (using thermal grease)
T_{CMAX}	= 90°C	Case temperature

- Estimate the total power that needs to be dissipated by the heat sink, Ps:

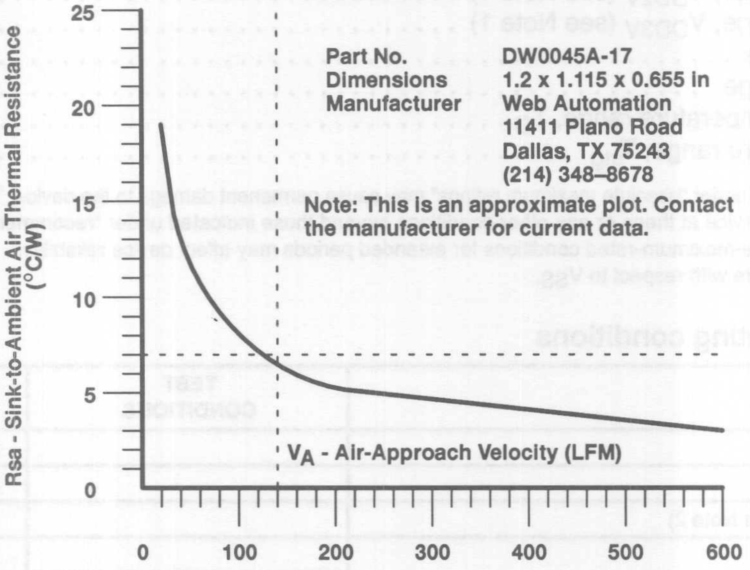
Ps	= Pa – 2 W	Assume 2W dissipation through the board
Ps	= 6.9 – 2 = 4.9 W	

- Compute the maximum heat sink-to-air thermal resistance RΘsa that maintains the case temperature below T_{CMAX} for the operating conditions specified previously:

RΘsa + RΘcs = $(T_C – T_A)$ / Ps
RΘsa = {$T_C – T_A$) / Ps} – RΘcs
RΘsa = {(90 – 55) / 4.9} – 0.2
RΘsa = 6.9°C/W

- Choose the heat sink based on the computed maximum sink-to-air resistance RΘsa and the minimum air approach velocity inside the enclosure. The heat-sink performance plot in Figure 21 verifies that at 85 LFM air velocity, the sink-to-air thermal resistance indeed stays at, or below, the Rsa value of 6.9°C/W that is required to maintain the case temperature below T_{CMAX} for the listed operating conditions. Actual measurement of the case temperature must be performed next to verify heat sink performance.

heat sink selection process – example 2 (continued)



Part No.	**DW0045A-17**
Dimensions	**1.2 x 1.115 x 0.655 in**
Manufacturer	**Web Automation** **11411 Plano Road** **Dallas, TX 75243** **(214) 348–8678**

Note: This is an approximate plot. Contact the manufacturer for current data.

Figure 21. Example 2 Heat-Sink Performance Plot

absolute maximum ratings over operating case temperature range (unless otherwise noted)†

Supply voltage range, V_{DD2V} (see Note 1)	–0.3 V to 3 V
Supply voltage range, V_{DD3V} (see Note 1)	–0.3 V to 4 V
Input voltage range	–0.3 V to 4 V
Output voltage range	–0.3 V to 4 V
Operating case temperature range, T_C	0°C to 90°C
Storage temperature range, T_{stg}	–55°C to 150°C

† Stresses beyond those listed under “absolute maximum ratings” may cause permanent damage to the device. These are stress ratings only, and functional operation of the device at these or any other conditions beyond those indicated under “recommended operating conditions” is not implied. Exposure to absolute-maximum-rated conditions for extended periods may affect device reliability.

NOTE 1: All voltage values are with respect to V_{SS}.

recommended operating conditions

		TEST CONDITIONS	MIN	NOM	MAX	UNIT
V_{DD2V}	Supply voltage		2.38	2.5	2.62	V
V_{DD3V}	Supply voltage		3.14	3.3	3.46	V
V_{SS}	Supply voltage (see Note 2)		0	0	0	V
V_{IH}	High-level input voltage		2.0			V
V_{IL}	Low-level input voltage				0.8	V
I_{OH}	High-level output current				– 12	mA
I_{OL}	Low-level output current				12	mA
T_C	Case temperature		0		90	°C

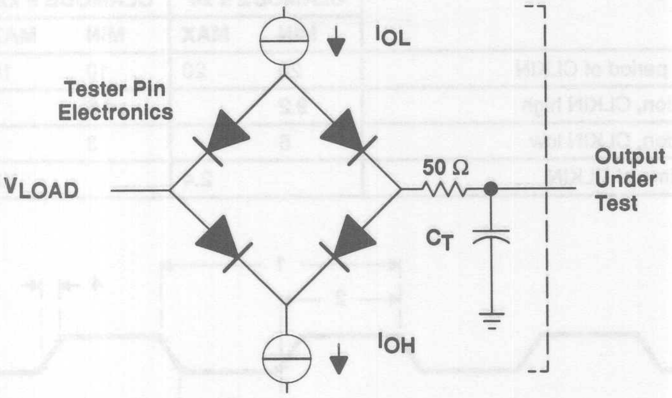
NOTE 2: In order to minimize noise on V_{SS}, care should be taken to provide a minimum inductance path between the V_{SS} pins and system ground.

electrical characteristics over recommended ranges of supply voltage and operating case temperature

PARAMETER		TEST CONDITIONS		MIN	TYP	MAX	UNIT
V_{OH}	High-level output voltage	V_{DD3V} = MIN	I_{OH} = MAX	2.4			V
V_{OL}	Low-level output voltage	V_{DD3V} = MIN	I_{OL} = MAX			0.6	V
I_I	Input current	$V_I = V_{SS}$ to V_{DD3V}				± 10	µA
I_{OZ}	Off-state output current	$V_O = V_{DD3V}$ or 0 V				± 10	µA
I_{DD2V}	Supply current	V_{DD2V} = MAX	200 MHz		2.5‡	TBD	A
I_{DD3V}	Supply current	V_{DD3V} = MAX	200 MHz		258	TBD	mA
C_I	Input capacitance				10		pF
C_O	Output capacitance				10		pF

‡ I_{DD2V} parameter is reduced in future revisions of the ’320C6201.

PARAMETER MEASUREMENT INFORMATION



Where:
I_{OL} = 12 mA (all outputs)
I_{OH} = – 12 mA (all outputs)
V_{LOAD} = 2.1 V
C_T = 30-pF typical distributed load circuit capacitance

Figure 22. Test Load Circuit

signal transition levels

TTL-output levels are driven to a minimum logic-high level of 2.4 V and to a maximum logic-low level of 0.6 V. Figure 23 shows the TTL-level outputs.

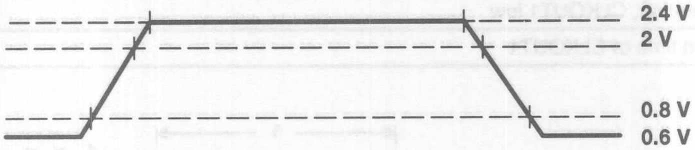


Figure 23. TTL-Level Outputs

TTL-output transition times are specified as follows:

- For a *high-to-low transition*, the level at which the output is said to be no longer high is 2 V, and the level at which the output is said to be low is 0.8 V.
- For a *low-to-high transition*, the level at which the output is said to be no longer low is 0.8 V, and the level at which the output is said to be high is 2 V.

Figure 24 shows the TTL-level inputs.

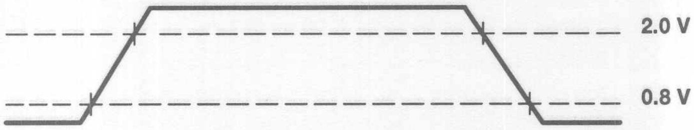


Figure 24. TTL-Level Inputs

TTL-compatible input transition times are specified as follows:

- For a *high-to-low transition* on an input signal, the level at which the input is said to be no longer high is 2 V, and the level at which the input is said to be low is 0.8 V.
- For a *low-to-high transition* on an input signal, the level at which the input is said to be no longer low is 0.8 V, and the level at which the input is said to be high is 2 V.

CLKIN timing requirements

NO.			CLKMODE = x4		CLKMODE = x2		CLKMODE = x1		unit
			MIN	MAX	MIN	MAX	MIN	MAX	
1	$t_{c(CLKIN)}$	Cycle time, period of CLKIN	20	20	10	10	5		ns
2	$t_{w(CLKINH)}$	Pulse duration, CLKIN high	9.2		4.6		1.9		ns
3	$t_{w(CLKINL)}$	Pulse duration, CLKIN low	6		3		1.9		ns
4	$t_{T(CLKIN)}$	Transition time of CLKIN		2.4		1.2		0.6	ns

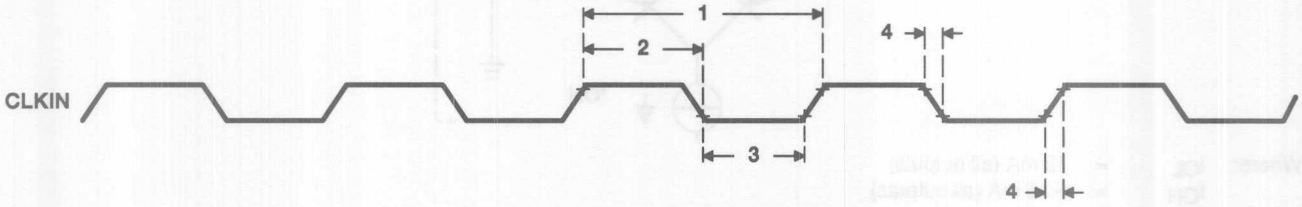


Figure 25. CLKIN Timing

CLKOUT1 timing requirements

NO.			MIN	MAX	UNIT
1	$t_{c(CKO1)}$	Cycle time, period of CLKOUT1	5		ns
2	$t_{w(CKO1H)}$	Pulse duration, CLKOUT1 high	1.9		ns
3	$t_{w(CKO1L)}$	Pulse duration, CLKOUT1 low	1.9		ns
4	$t_{T(CKO1)}$	Transition time of CLKOUT1		0.6	ns

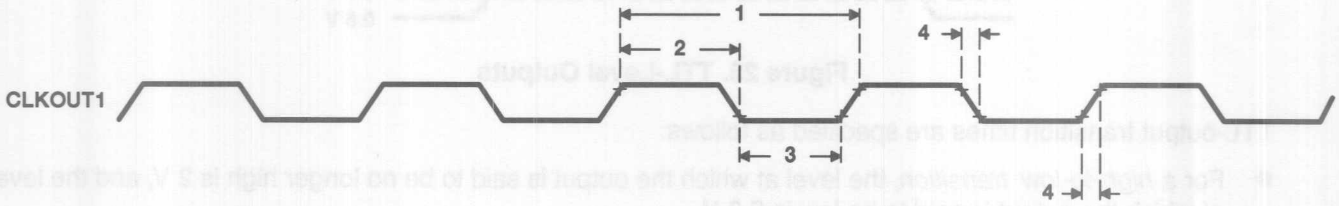


Figure 26. CLKOUT1 Timing

CLKOUT2 timing requirements

NO.			MIN	MAX	UNIT
1	$t_{c(CKO2)}$	Cycle time, period of CLKOUT2	10		ns
2	$t_{w(CKO2H)}$	Pulse duration, CLKOUT2 high	3.8		ns
3	$t_{w(CKO2H)}$	Pulse duration, CLKOUT2 low	3.8		ns
4	$t_{T(CKO2)}$	Transition time of CLKOUT2		0.6	ns

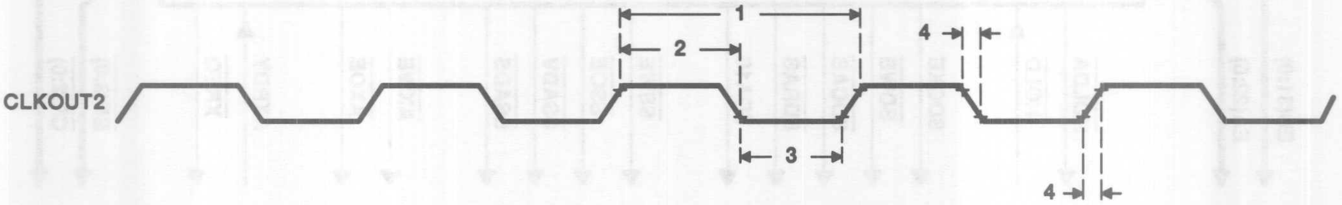


Figure 27. CLKOUT2 Timing

external memory interface

The external memory interface (EMIF) accesses data that is external to the chip. As shown in Figure 28, the EMIF can receive requests for external data access from one of three functional blocks – the data memory controller, the program/cache controller, or the DMA controller. The EMIF operation is configurable through memory-mapped control registers accessed by the internal peripheral bus. See Table 5 and Figure 7. Every EMIF program or data access uses the common 23-bit address bus and 32-bit data bus.

There are three spaces in the memory map, each of which is represented by one of three chip-enable signals – $\overline{\text{CE2}}$, $\overline{\text{CE1}}$, and $\overline{\text{CE0}}$. One of the chip-enables has to be asserted when the processors reads or writes to any of the three memory-map external spaces 2, 1, or 0. Byte-enable control signals, $\overline{\text{BE3}}$–$\overline{\text{BE0}}$, select the individual bytes, half-words, or words during EMIF-write cycles that originate from the data-memory controller. All program and DMA cycles, as well as data-controller read cycles, are 32-bit – resulting in all four $\overline{\text{BE}}$ signals being active. Data controller write cycles, however, can access byte or half-word data using individual $\overline{\text{BE}}$ controls to select the active bytes. The addressing of bytes within each word is set by the LENDIAN pin to be in the direction of high bits or low bits.

$\overline{\text{CE}}$ spaces 0 and 2 can be programmed to support different types of memory, but $\overline{\text{CE}}$ space 1 is restricted to asynchronous memory cycles only, including ROM typically used to boot the system at power up. The ROM read cycles are identical to $\overline{\text{CE0}}$ and $\overline{\text{CE2}}$ asynchronous SRAM cycles from the control-signal perspective, although the $\overline{\text{CE1}}$ cycles have the flexibility to pack ROM bytes into words if the ROM is 8- or 16-bits wide. The external ROM size must be encoded by way of pins DC7–DC6 during reset.

The types of memory assigned to $\overline{\text{CE}}$ spaces 0 and 2 are also encoded by way of the DC pins during reset. Pins DC3–DC2 specify the type of memory for the $\overline{\text{CE0}}$ space and pins DC5–DC4 specify the type of memory for space $\overline{\text{CE2}}$. The supported memory types include asynchronous memory, synchronous burst SRAM (SBSRAM), and synchronous DRAM (SDRAM). An additional external-control cycle can be used to drive $\overline{\text{CE}}$ spaces 0 and 2 by setting appropriate bits in the EMIF global control register. External control cycles are different from memory cycles in that they use request/ready handshaking to stretch individual bus cycles to accommodate longer response times inherent in slower I/O devices. The EMIF provides a separate set of control signals for each one of the memory cycles listed above (see Figure 28).

EMIF $\overline{\text{HOLD}}$ and $\overline{\text{HOLDA}}$ signals arbitrate the ownership of the EMIF bus between the host and the TMX320C6201 DSP.

external memory interface (continued)

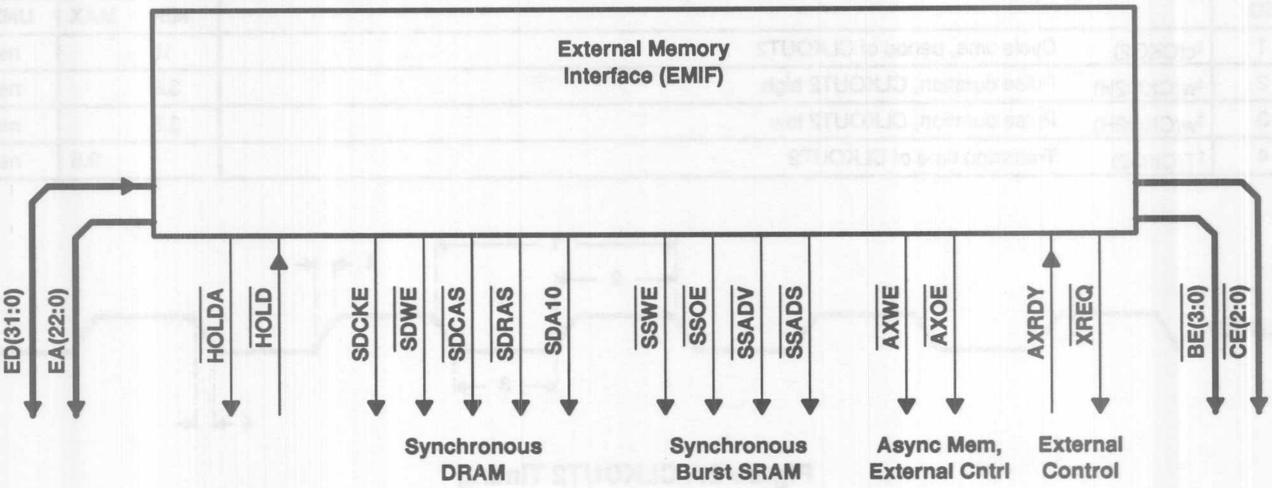


Figure 28. External Memory Interface

asynchronous memory cycles

Asynchronous memory read cycles use the $\overline{\text{AXOE}}$ control signal to enable the memory device to place the data on the data bus. EMIF CE-space timing registers define the shape of the read cycle and the amount of time that each address is valid. The shape of the read cycle in Figure 29, reflects the fastest possible asynchronous read corresponding to one-cycle setup, one-cycle strobe width, and one-cycle hold fields of the EMIF CE space timing register.

Asynchronous memory write cycles use the $\overline{\text{AXWE}}$ control signal to latch the data in the memory device. The $\overline{\text{AXWE}}$ signal shape is programmed by setting fields in the EMIF CE-space timing register. The cycles in Figure 30 correspond to one-cycle set up, one-cycle strobe width, and one-cycle hold time (fastest write possible). Asynchronous memory $\overline{\text{AXOE}}$ and $\overline{\text{AXWE}}$ control signals are also used during external control cycles.

timing requirements for asynchronous memory cycles

NO			MIN	MAX	UNIT
5	$t_{su(D-CKH)}$	Set-up time, read D before CLKOUT1 high	0.5		ns
6	$t_{h(CKH-D)}$	Hold time, read D valid after CLKOUT1 high	0		ns

switching characteristics for asynchronous memory cycles

NO	PARAMETER		TEST CONDITIONS	MIN	MAX	UNIT
1	$t_{d(CKH-CE)}$	Delay time, CLKOUT1 high to $\overline{\text{CE}}$ valid			1.9	ns
2	$t_{d(CKH-BE)}$	Delay time, CLKOUT1 high to $\overline{\text{BE}}$ valid			2.2	ns
3	$t_{d(CKH-A)}$	Delay time, CLKOUT1 high to EA valid		0†	1.7	ns
4	$t_{d(CKH-D)}$	Delay time, CLKOUT1 high to ED valid		0†	2.0	ns
7	$t_{d(CKH-OE)}$	Delay time, CLKOUT1 high to $\overline{\text{AXOE}}$ valid			1.9	ns
8	$t_{d(CKH-WE)}$	Delay time, CLKOUT1 high to $\overline{\text{AXWE}}$ valid			2.0	ns

† Values specified by design and not tested

PARAMETER MEASUREMENT INFORMATION

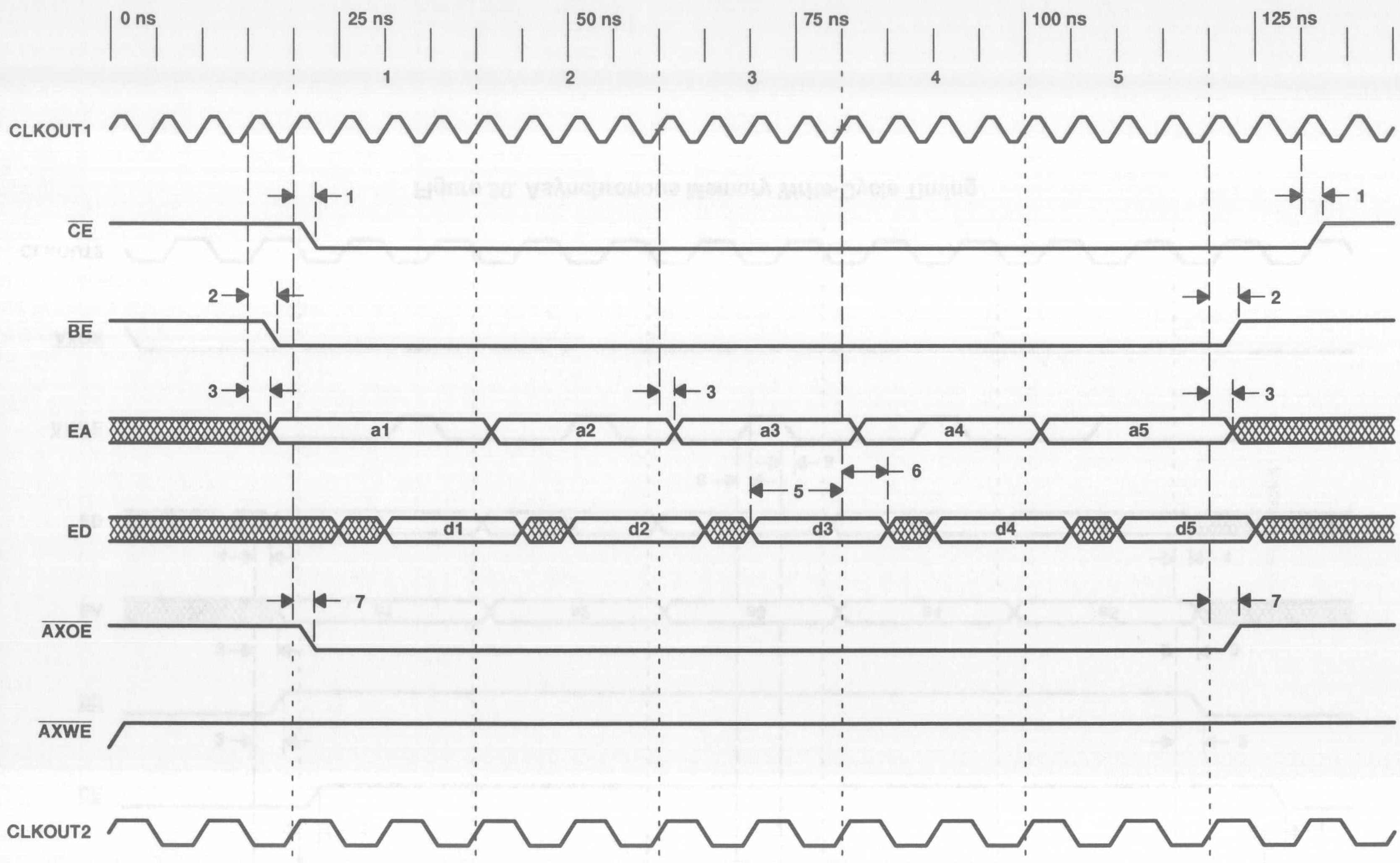


Figure 29. Asynchronous Memory Read-Cycle Timing

PRODUCT PREVIEW

PARAMETER MEASUREMENT INFORMATION

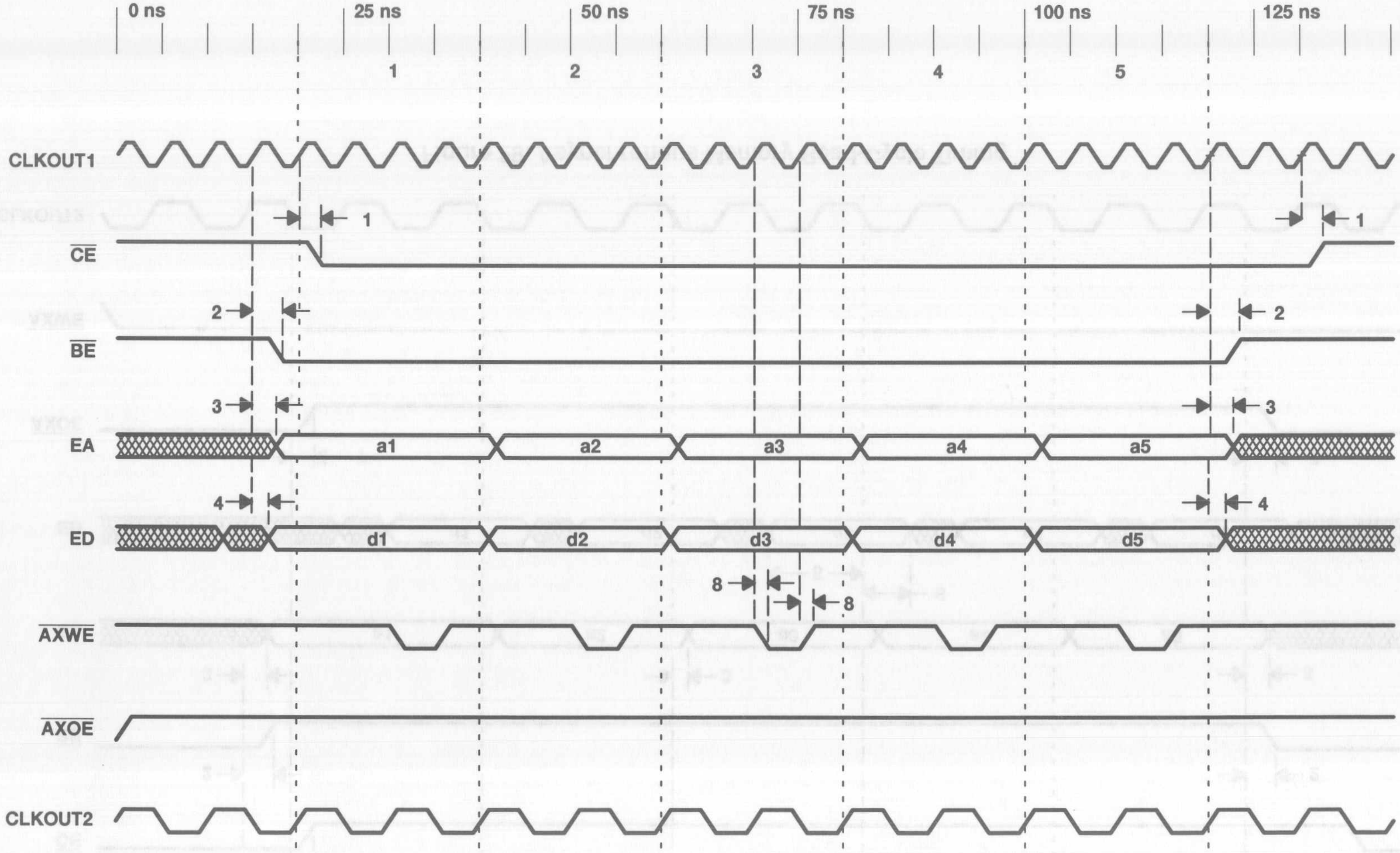


Figure 30. Asynchronous Memory Write-Cycle Timing

asynchronous memory cycles

The CE space timing register contains three fields that determine the shape and duration of the $\overline{\text{AXWE}}$ and $\overline{\text{AXOE}}$ control signals. The examples shown in Figure 29 and Figure 30 correspond to value of 1 programmed in set-up, strobe, and hold fields of the CE space timing register (the smallest value permitted). Each read and write also contains a one-cycle internal address latch time which is not programmable. Even as the $\overline{\text{AXOE}}$ signal does not come up in between the successive read cycles, the combined effect of set-up, strobe, and hold values determines the total length of each read (see Figure 31).

	31 20	19 16	15 14	13 8	7 2	1 0
CE Space Timing Register	RSV	SETUP†	RSV	STROBE†	RSV	HOLD†
	Reserved	Set-up time before strobe falling edge	Reserved	Active strobe width	Reserved	Hold time after strobe rising edge

† The smallest value permitted for this field is 1.

Figure 31. CE Space Timing Register

external controller cycles

Any external memory access in $\overline{\text{CE}}$ space 0 and 2 can be converted to an external controller cycle by setting appropriate bits in the EMIF global control register. All external controller cycles use $\overline{\text{AXOE}}$ and $\overline{\text{AXWE}}$ control signals to specify the direction of data transfer. For both reads and writes, the '320C6201 DSP drives the $\overline{\text{XREQ}}$ control signal low to announce the start of a read or write cycle. If the external device is ready for the data transfer, it immediately drives the AXRDY signal high. AXRDY remains high as long as the external device is ready to transfer data. When the external device is not ready to transfer, it immediately brings the AXRDY line low to stall the current bus cycle until it is ready again. See Figure 32 and Figure 33 for timing waveforms.

timing requirements for external controller cycles

NO			MIN	MAX	UNIT
5	$t_{su(D-CKH)}$	Setup time, read ED before CLKOUT1 high	1.0		ns
6	$t_{h(CKH-D)}$	Hold time, read ED valid after CLKOUT1 high	2.0		ns
9	$t_{d(CKH-XREQ)}$	Delay time, CLKOUT1 high to $\overline{\text{XREQ}}$ valid		1.3	ns

switching characteristics for external controller cycles

NO	PARAMETER		MIN	MAX	UNIT
1	$t_{d(CKH-CE)}$	Delay time, CLKOUT1 high to $\overline{\text{CE}}$ valid		1.9	ns
2	$t_{d(CKH-BE)}$	Delay time, CLKOUT1 high to $\overline{\text{BE}}$ valid		2.2	ns
3	$t_{d(CKH-A)}$	Delay time, CLKOUT1 high to EA valid	0‡	1.7	ns
4	$t_{d(CKH-D)}$	Delay time, CLKOUT1 high to ED valid	0‡	2.0	ns
7	$t_{d(CKH-OE)}$	Delay time, CLKOUT1 high to $\overline{\text{AXOE}}$ valid		1.9	ns
8	$t_{d(CKH-WE)}$	Delay time, CLKOUT1 high to $\overline{\text{AXWE}}$ valid		2.0	ns
10	$t_{su(XRDY-CKH)}$	Setup time, read XRDY valid before CLKOUT1 high	0.5		ns
11	$t_{h(CKH-XRDY)}$	Hold time, read XRDY valid after CLKOUT1 high	2.8		ns

‡ Values specified by design and not tested

PRODUCT PREVIEW

PARAMETER MEASUREMENT INFORMATION

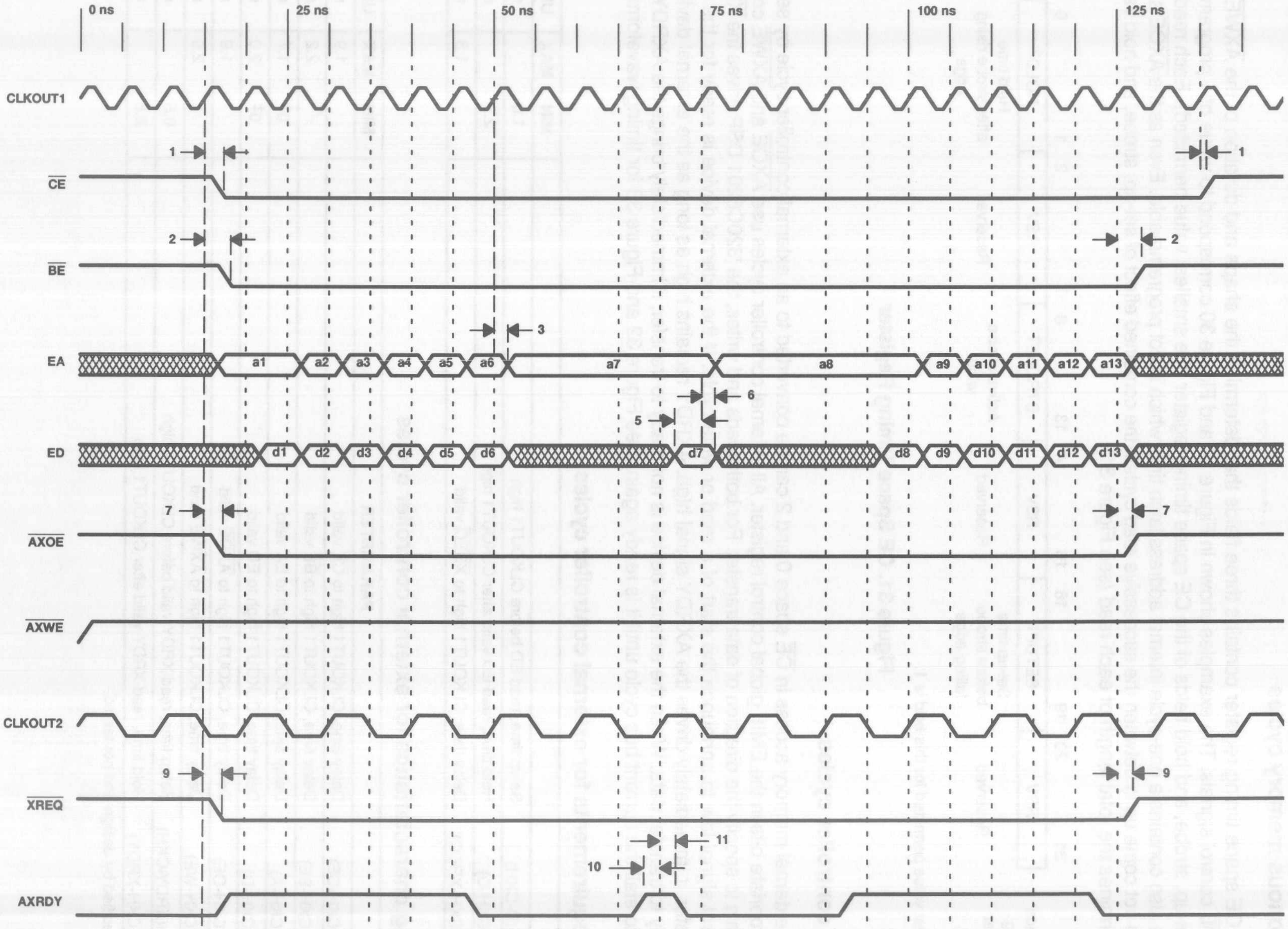


Figure 32. External Controller Read-Cycle Timing

PARAMETER MEASUREMENT INFORMATION

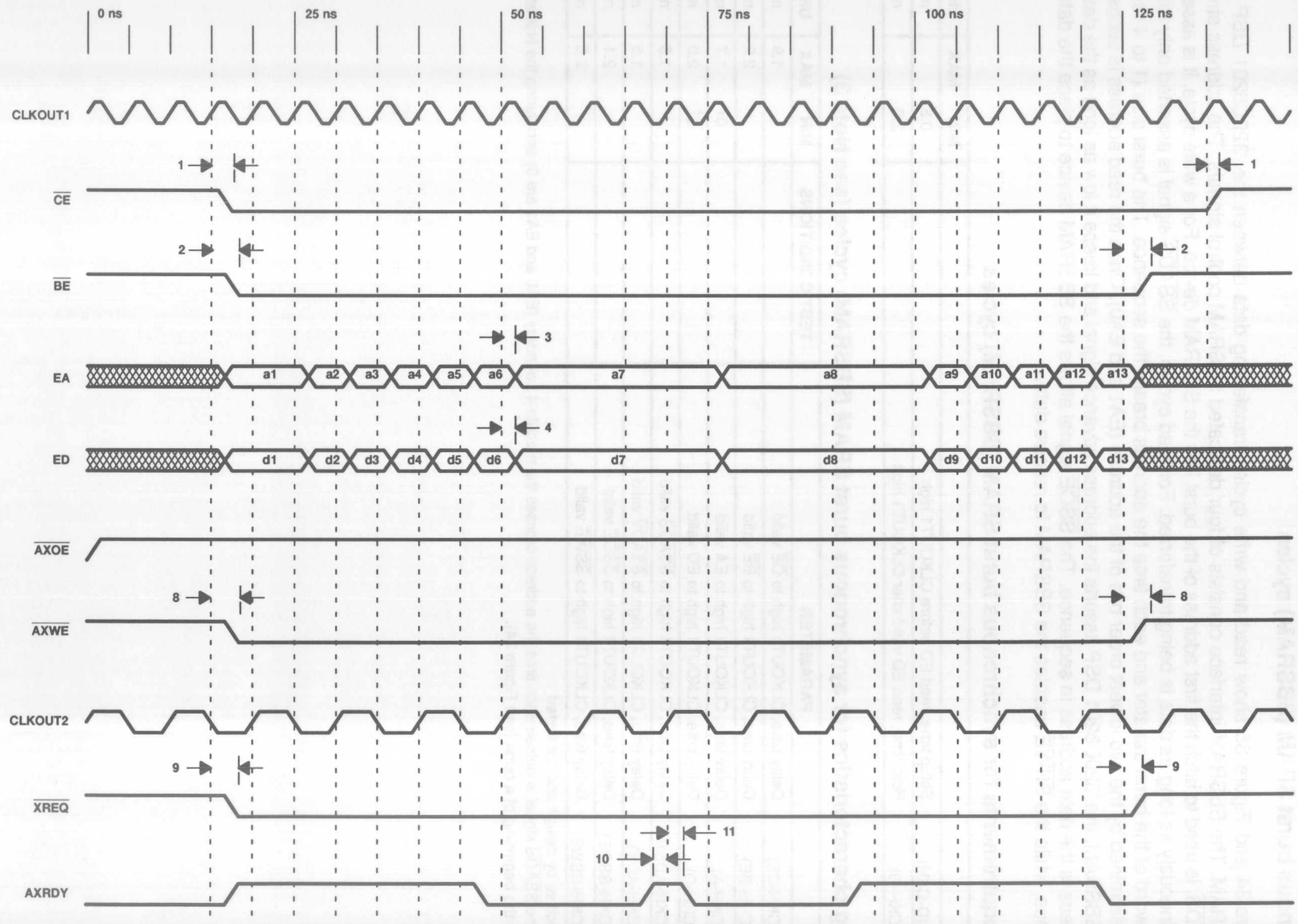


Figure 33. External Controller Write-Cycle Timing

synchronous burst SRAM (SBSRAM) cycles

Figure 34 and Figure 35 show read and write cycles transferring data between the ’320C6201 DSP and SBSRAM. The SBSRAM interface consists of four dedicated SBSRAM control signals. The address strobe, $\overline{\text{SSADS}}$, is used to latch the first address of the burst into the SBSRAM device. For a write cycle, it is asserted continuously as long as data is being transferred. For read cycles, the $\overline{\text{SSADS}}$ signal is asserted only for the first word of the burst transfer and each time the address breaks the sequence. The burst size (1 to 4 reads) is determined by the two lowest order bits of the address (EA1 and EA0). Once the read address is latched in the SBSRAM, the ’320C6201 DSP asserts the address advance signal and keeps it low as long as the desired address is the next address in sequence. The $\overline{\text{SSOE}}$ signal allows the SBSRAM device to place the data on the bus, while the $\overline{\text{SSWE}}$ enables the SBSRAM to receive data.

timing requirements for synchronous burst SRAM (SBSRAM) cycles

NO			MIN	MAX	UNIT
5	$t_{su(D\text{-}CKH)}$	Setup time, read ED before CLKOUT1 high	0.5		ns
6	$t_{h(CKH\text{-}D)}$	Hold time, read ED valid after CLKOUT1 high	2.0		ns

switching characteristics for synchronous burst SRAM (SBSRAM) cycles (see Note 3)

NO	PARAMETER		TEST CONDITIONS	MIN	MAX	UNIT
1	$t_{d(CKH\text{-}CE)}$	Delay time, CLKOUT1 high to $\overline{\text{CE}}$ valid			1.9	ns
2	$t_{d(CKH\text{-}BE)}$	Delay time, CLKOUT1 high to $\overline{\text{BE}}$ valid			2.2	ns
3	$t_{d(CKH\text{-}A)}$	Delay time, CLKOUT1 high to EA valid		0†	1.7	ns
4	$t_{d(CKH\text{-}D)}$	Delay time, CLKOUT1 high to ED valid		0†	2.0	ns
7	$t_{d(CKH\text{-}ADS)}$	Delay time, CLKOUT1 high to $\overline{\text{SSADS}}$ valid			1.5	ns
8	$t_{d(CKH\text{-}ADV)}$	Delay time, CLKOUT1 high to $\overline{\text{SSADV}}$ valid			1.8	ns
9	$t_{d(CKH\text{-}SSOE)}$	Delay time, CLKOUT1 high to $\overline{\text{SSOE}}$ valid			2.1	ns
10	$t_{d(CKH\text{-}SSWE)}$	Delay time, CLKOUT1 high to $\overline{\text{SSWE}}$ valid			2.0	ns

† Values specified by design and not tested

NOTE 3: The $\overline{\text{SSADS}}$ signal is pulsed each time the address crosses the modulo 4 boundary (EA0 and EA1 are 0) and during the first access at the beginning of a burst (see Figure 34).

PARAMETER MEASUREMENT INFORMATION

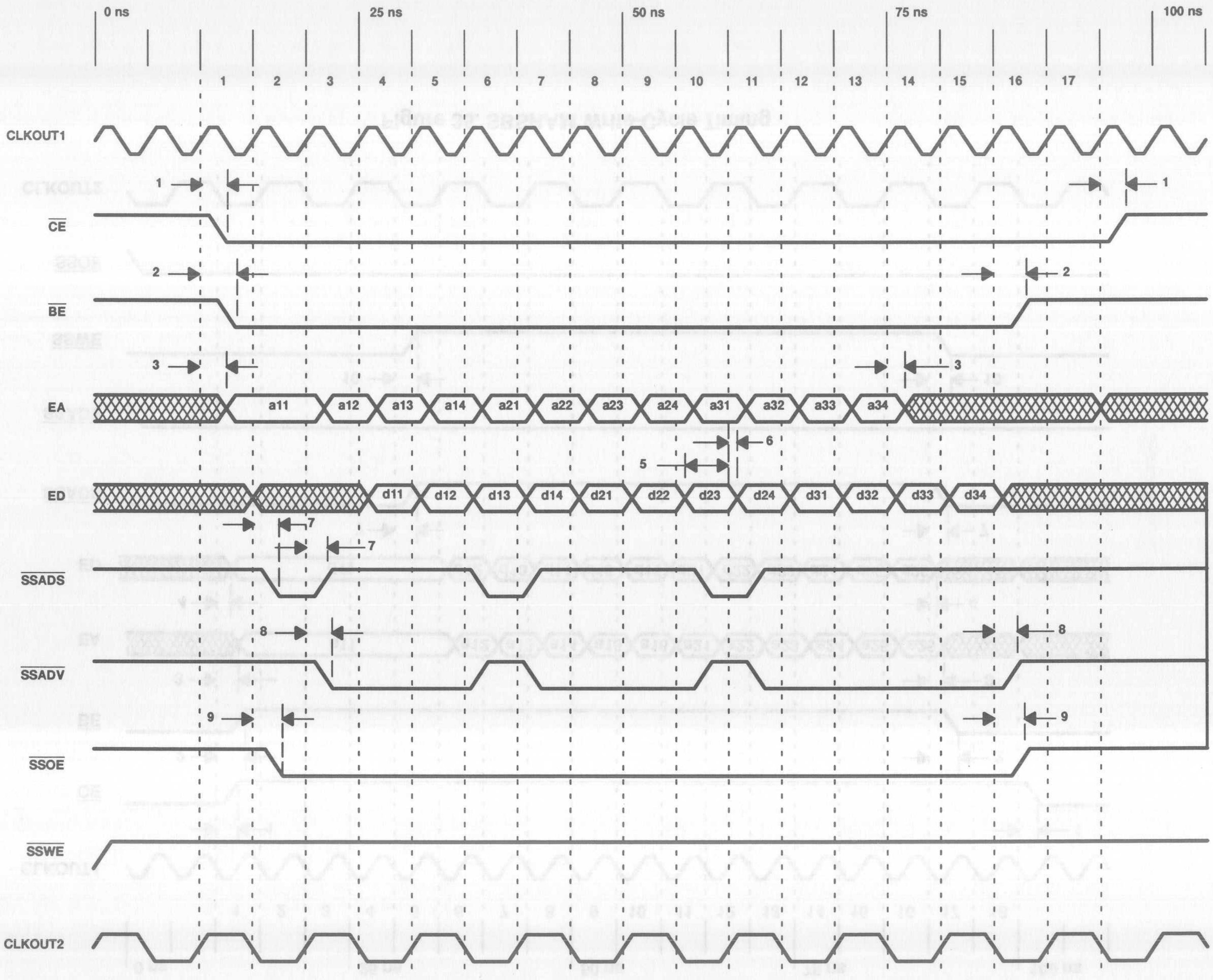


Figure 34. SBSRAM Read-Cycle Timing

PRODUCT PREVIEW

PRODUCT PREVIEW

PARAMETER MEASUREMENT INFORMATION

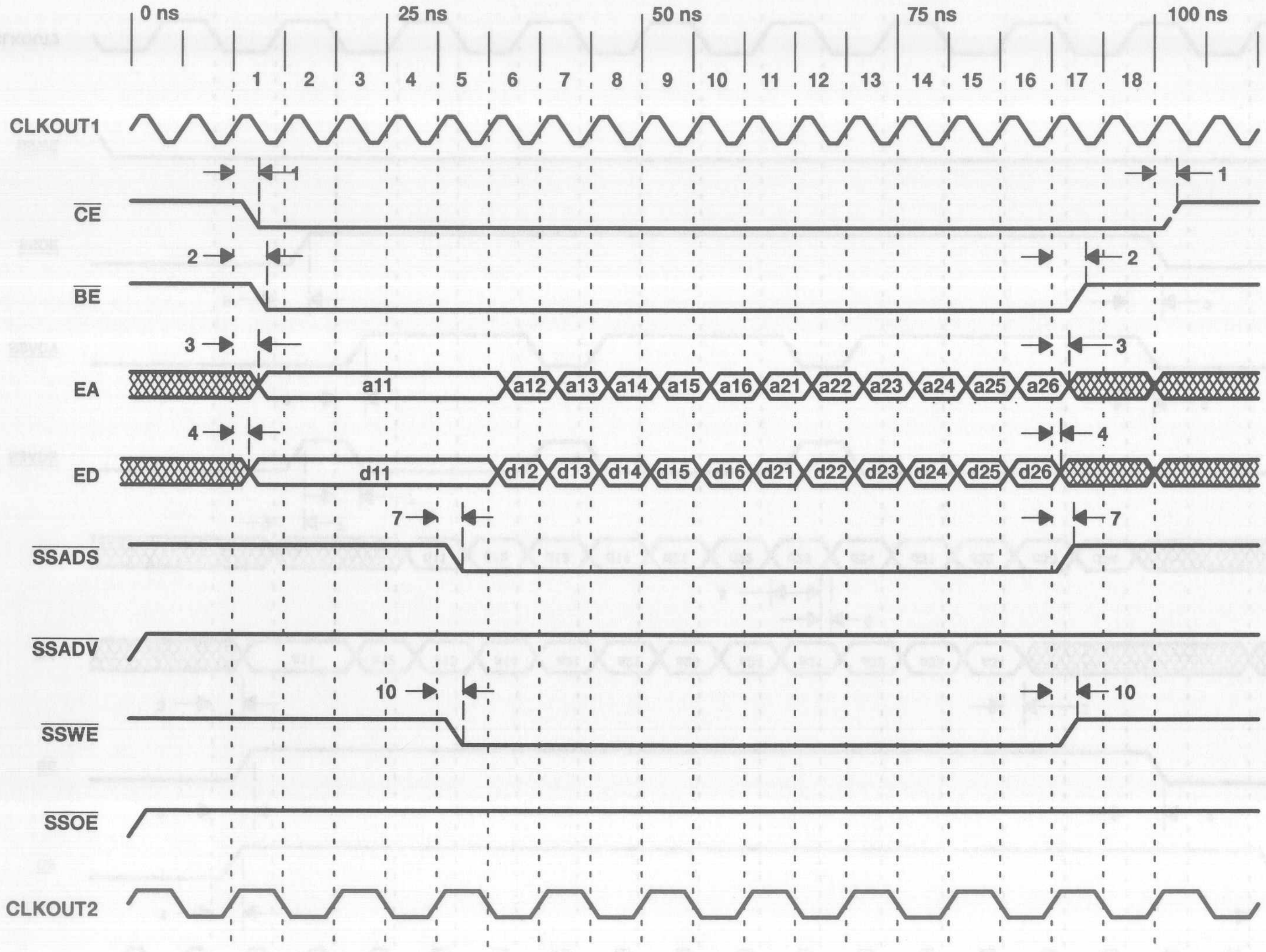


Figure 35. SBSRAM Write-Cycle Timing

synchronous DRAM (SDRAM) cycles

The SDRAM cycles on the ‘C6201 device are limited to non-burst operation for the initial version of the ’320C6201. Both read and write cycles consist of three commands sent sequentially to the SDRAM device (see Figure 36 and Figure 37). Each of the commands is encoded by ’320C6201 by a different mix of signal levels on $\overline{\text{CE}}$, SDA10, $\overline{\text{SDRAS}}$, $\overline{\text{SDCAS}}$, and $\overline{\text{SDWE}}$ control lines. First, the ACTV command activates one of two SDRAM internal banks. The READ or WRITE commands transfer data between the ’320C6201 device and the SDRAM memory. Finally, the DCAB command deactivates both banks of the SDRAM (banks A and B). The SDRAM uses the CLKOUT2 clock to transfer data and to latch commands. In a read cycle, the data is driven on the bus by the SDRAM exactly two CLKOUT2 cycles following the rising edge of $\overline{\text{SDCAS}}$. The ’320C6201 DSP programs read latency into the SDRAM at the time of SDRAM initialization.

The $\overline{\text{BE}}$ control signals behave differently during SDRAM cycles than they do for other types of memory. Each $\overline{\text{BE}}$ signal must be connected to the corresponding SDRAM DQM input. For reads, the DQM controls become output-enables and for the writes they assume a function of write masking (associated with burst operation of SDRAM).

Figure 38 shows the initialization commands sent to the SDRAM following reset. First, both banks of the SDRAM are deactivated with the DCAB command. Next, eight refresh cycles are issued, followed by the mode register set (MRS) command that initializes control registers inside the SDRAM in accordance with the values stored in the EMIF SDRAM control register. During the MRS command, the SDRAM initialization data is actually output on the address bus instead of on the data bus.

The SDRAM device must be refreshed periodically. Since one of the address pins, EA10, is involved in the refresh operation, it has been duplicated for the exclusive use of the SDRAM interface to avoid interfering with other memory cycles that can be active during refresh. Figure 39 shows an SDRAM-refresh cycle in the middle of an asynchronous-SRAM-read sequence. Two commands are used in a single refresh cycle. The first one deactivates both SDRAM banks (DCAB) and the second one performs a $\overline{\text{CAS}}$-before-$\overline{\text{RAS}}$ refresh. It is evident in the figure that despite the use of address lead EA10 by the DCAB command, the asynchronous memory read cycles are progressing uninterrupted due to the duplicate SDA10 pin.

timing requirements for synchronous DRAM (SDRAM) cycles

NO			MIN	MAX	UNIT
5	$t_{su(D\text{-}CKH)}$	Setup time, read ED before CLKOUT1 high	0.5		ns
6	$t_{h(CKH\text{-}D)}$	Hold time, read ED valid after CLKOUT1 high	2.0		ns

switching characteristics for synchronous DRAM (SDRAM) cycles

NO	PARAMETER		TEST CONDITIONS	MIN	MAX	UNIT
1	$t_{d(CKH\text{-}CE)}$	Delay time, CLKOUT1 high to $\overline{\text{CE}}$ valid			1.9	ns
3	$t_{d(CKH\text{-}A)}$	Delay time, CLKOUT1 high to EA valid		0†	1.7	ns
4	$t_{d(CKH\text{-}D)}$	Delay time, CLKOUT1 high to ED valid		0†	2.0	ns
7	$t_{d(CKH\text{-}SDA10)}$	Delay time, CLKOUT1 high to $\overline{\text{SDA10}}$ valid			1.5	ns
8	$t_{d(CKH\text{-}SDRAS)}$	Delay time, CLKOUT1 high to $\overline{\text{SDRAS}}$ valid			1.1	ns
9	$t_{d(CKH\text{-}SDCAS)}$	Delay time, CLKOUT1 high to $\overline{\text{SDCAS}}$ valid			1.3	ns
10	$t_{d(CKH\text{-}SDWE)}$	Delay time, CLKOUT1 high to $\overline{\text{SDWE}}$ valid			1.1	ns
11	$t_{d(CKH\text{-}SDCKE)}$	Delay time, CLKOUT1 high to SDCKE			1.3	ns

† Values specified by design and not tested

PRODUCT PREVIEW

PRODUCT PREVIEW

PARAMETER MEASUREMENT INFORMATION

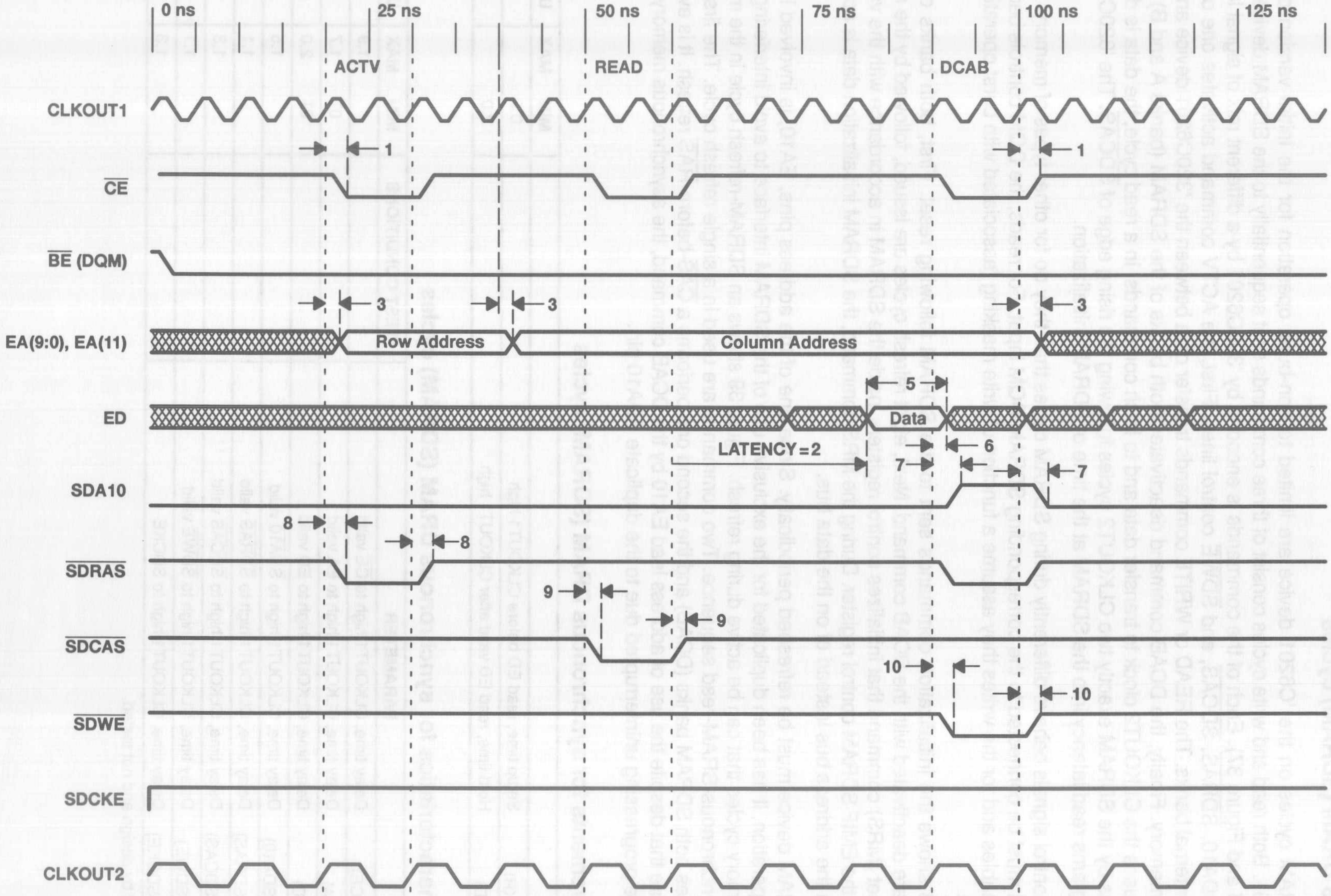


Figure 36. SDRAM Read-Cycle Timing

PARAMETER MEASUREMENT INFORMATION

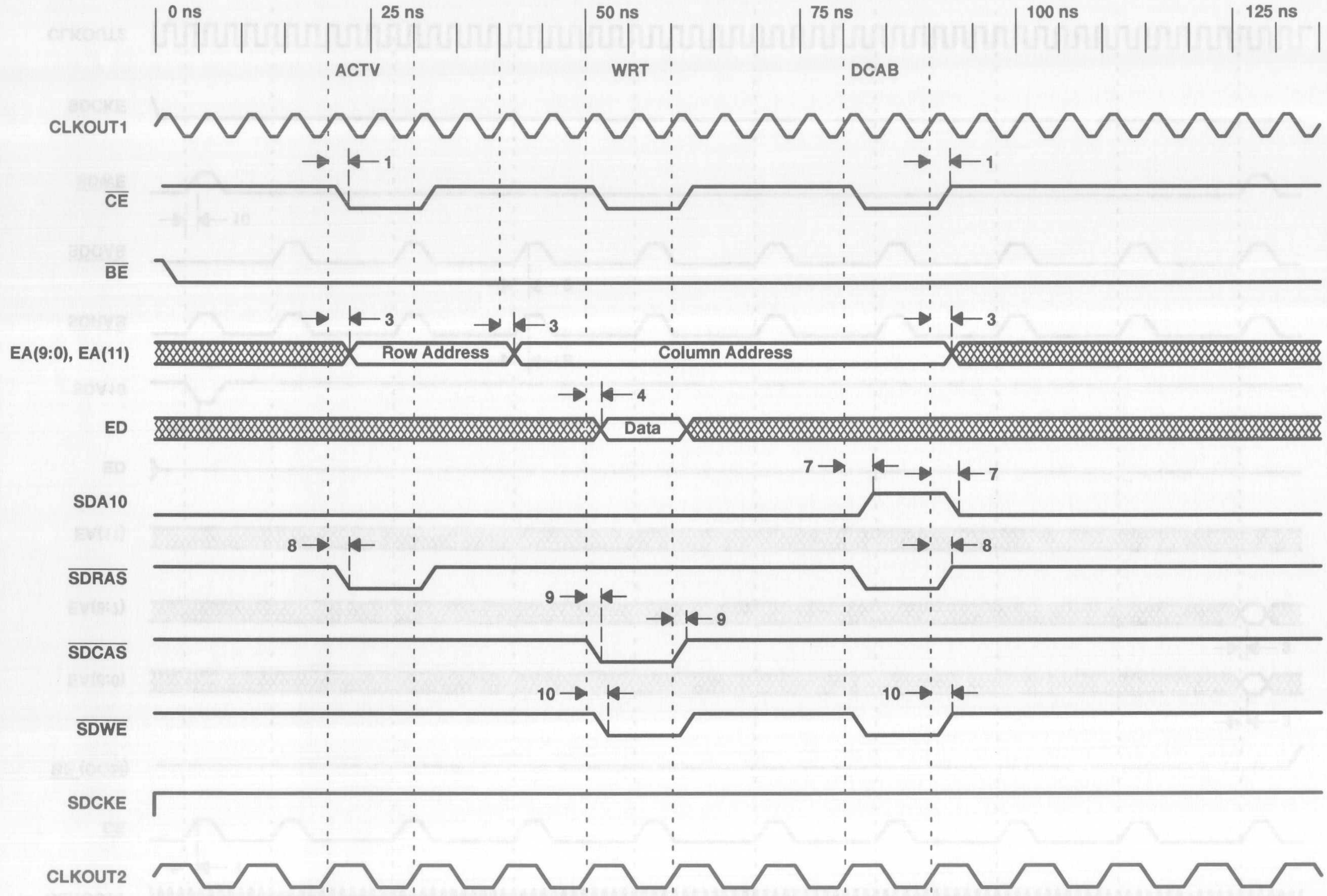


Figure 37. SDRAM Write-Cycle Timing

PARAMETER MEASUREMENT INFORMATION

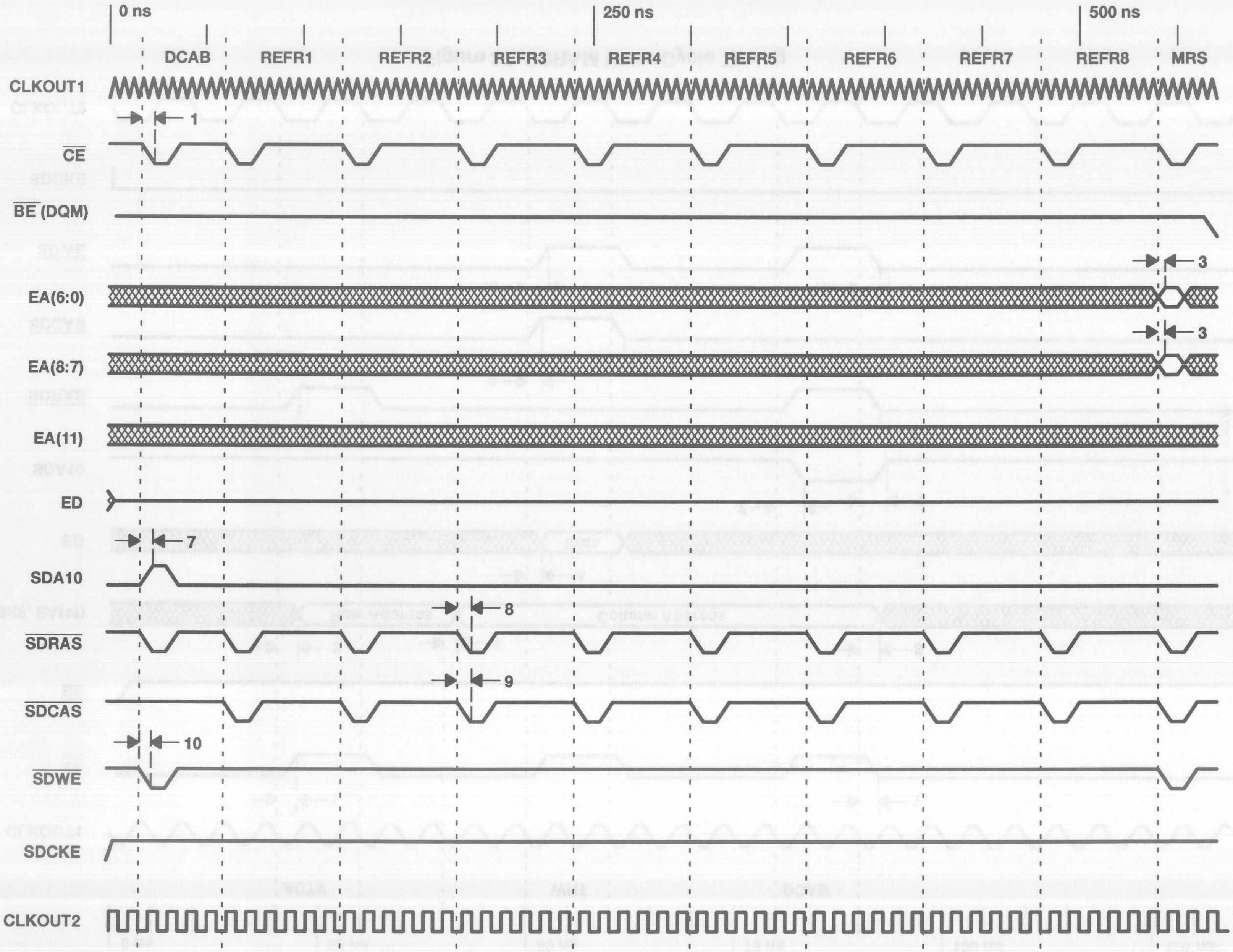


Figure 38. SDRAM Initialization-Cycle Timing

PARAMETER MEASUREMENT INFORMATION

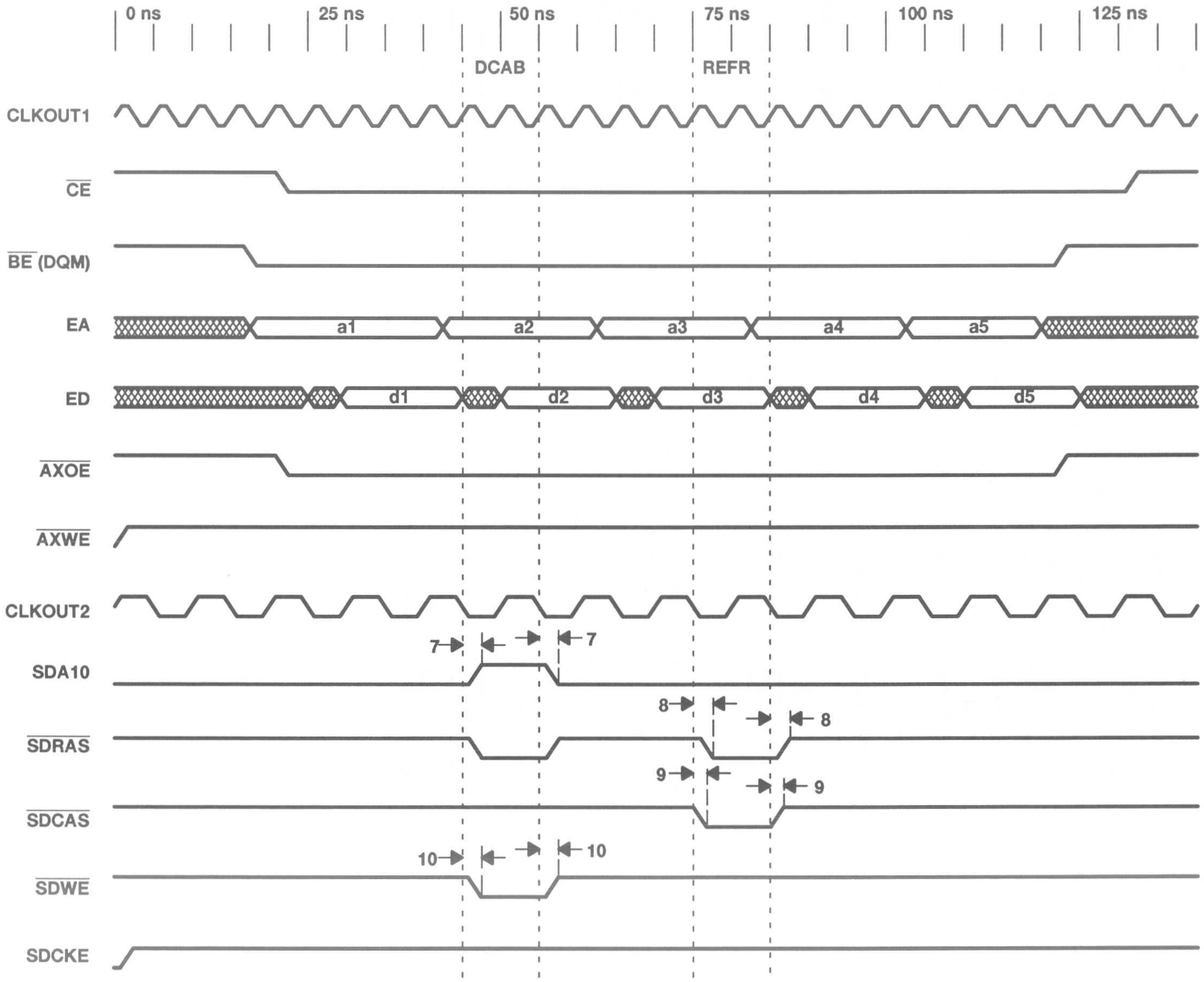


Figure 39. SDRAM Refresh-Cycle Timing

PRODUCT PREVIEW

interrupt response cycles

An external-interrupt event is triggered once by a high pulse on an INTx signal. If, on one rising edge of the CLKOUT1, the INTx is recognized as low and on the next CLKOUT1 rising edge the INTx is recognized as high, an external interrupt is detected. As early as three cycles later, the IACK signal can pulse for one cycle with the number of the active interrupt encoded on the four INUM lines. The IACK and INUM signals indicate the start of any interrupt processing, not just for external interrupts. The INUM identification number reflects the relative bit position of the processed interrupt within the interrupt flag register (IFR). Also, one cycle prior to the IACK pulse, all instructions (except branch delay slots, which delay IACK) that have not reached the execution stage are annulled and the address of the annulled instruction closest to execution stage is saved in the appropriate interrupt return pointer (IRP or NRP). The instructions that already have begun executing complete the remaining execution stages while the reset service fetch packet is being fetched and decoded. During the seven fetch/decode cycles the processor does not start processing any new instructions (hence the seven cycle interrupt overhead). The interrupt service fetch packet starts executing on the seventh cycle following IACK (see Figure 40).

timing requirements for the interrupt response cycle

NO			MIN	MAX	UNIT
4	$t_{w(INTR)}$	Pulse duration, width of the interrupt pulse	1		cycle
5	$t_{w(ILOW)}$	Pulse duration, width of the low state preceding interrupt pulse	1		cycle
6	$t_{su(IHIGH-CKH)}$	Setup time, $\overline{\text{INTx}}$ low before CLKOUT1 high	0.3		ns

switching characteristics for the interrupt response cycle

NO	PARAMETER		TEST CONDITIONS	MIN	MAX	UNIT
1	$t_{r(IACK)}$	Response time, IACK high after $\overline{\text{INTx}}$ low		3		cycles
2	$t_{w(IDEAD)}$	Pulse duration, width of the interrupt overhead (no instructions executed)		7	7	cycles
3	$t_{r(ISFP)}$	Response time, interrupt service fetch packet execution after $\overline{\text{INTX}}$ low		9		cycles
7	$t_{h(CKH-ILOW)}$	Hold time, $\overline{\text{INTx}}$ low after CLKOUT1 high		3.0		ns
8	$t_{d(CKH-IACK)}$	Delay time, CLKOUT1 high to IACK valid			1	ns
9	$t_{d(CKH-INUM)}$	Delay time, CLKOUT1 high to INUM valid			1	ns

PARAMETER MEASUREMENT INFORMATION

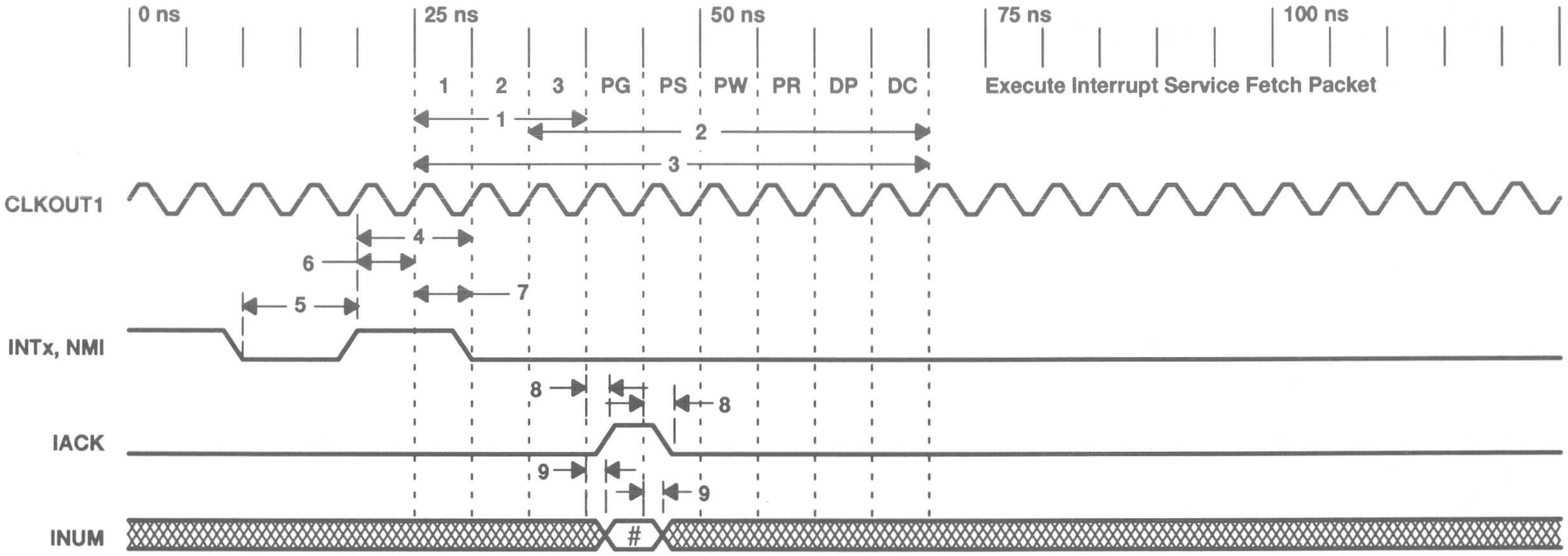


Figure 40. Interrupt-Response-Cycle Timing

reset timing

The $\overline{\text{RESET}}$ signal is sampled on every rising edge of CLKOUT1. If it is detected active (low), all EMIF output signals except HOLDA, as well as all host port interface HD signals are placed in the high-impedance mode on the following CLKOUT1 rising edge. At the same time, the HACK signal is driven to a low state. After the reset is deasserted, these signals in high-impedance mode can be driven again by the '320C6201 (see Figure 41).

Typically, the PLL locks up to the specified output frequency within tens of microseconds following device power up. The actual time depends on the PLL operating mode chosen and the CLKIN and CLKOUT1 frequency ranges.

Note that $\overline{\text{HOLD}}/\overline{\text{HOLDA}}$ logic is not affected by the state of the $\overline{\text{RESET}}$ signal.

timing requirements for the reset cycle

NO			MIN	MAX	UNIT
1	$t_{su(CKH\text{-}RESET)}$	Setup time, $\overline{\text{RESET}}$ before CLKOUT1 high	1		ns
2	$t_{w(RESET)}$	Pulse duration, $\overline{\text{RESET}}$ pulse (PLL stable)	10		cycles

PRODUCT PREVIEW

switching characteristics for the reset cycle

NO			TEST CONDITIONS	MIN	MAX	UNIT
3	$t_{v(CLKOUT)}$	Valid time, CLKOUT1/2 after $\overline{\text{RESET}}$ low (depends on PLLMODE) (see Note 4)		28	210	µs
5	$t_{d(CKH\text{-}CE)}$	Delay time, CLKOUT1 high to $\overline{\text{CE}}$ in high impedance mode/driven			5[†]	ns
6	$t_{d(CKH\text{-}BE)}$	Delay time, CLKOUT1 high to $\overline{\text{BE}}$ in high impedance mode/driven			5[†]	ns
7	$t_{d(CKH\text{-}A)}$	Delay time, CLKOUT1 high to EA in high impedance mode/driven		0[‡]	5[†]	ns
8	$t_{d(CKH\text{-}D)}$	Delay time, CLKOUT1 high to ED in high impedance mode/driven		0[‡]	5.1[†]	ns
9	$t_{d(CKH\text{-}XREQ)}$	Delay time, CLKOUT1 high to $\overline{\text{XREQ}}$ in high impedance mode/driven			5[†]	ns
10	$t_{d(CKH\text{-}OE)}$	Delay time, CLKOUT1 high to $\overline{\text{AXOE}}$ in high impedance mode/driven			5[†]	ns
11	$t_{d(CKH\text{-}WE)}$	Delay time, CLKOUT1 high to $\overline{\text{AXWE}}$ in high impedance mode/driven			5[†]	ns
12	$t_{d(CKH\text{-}ADS)}$	Delay time, CLKOUT1 high to $\overline{\text{SSADS}}$ in high impedance mode/driven			5[†]	ns
13	$t_{d(CKH\text{-}ADV)}$	Delay time, CLKOUT1 high to $\overline{\text{SSADV}}$ in high impedance mode/driven			5[†]	ns
14	$t_{d(CKH\text{-}SSOE)}$	Delay time, CLKOUT1 high to $\overline{\text{SSOE}}$ in high impedance mode/driven			5[†]	ns
15	$t_{d(CKH\text{-}SSWE)}$	Delay time, CLKOUT1 high to $\overline{\text{SSWE}}$ in high impedance mode/driven			5[†]	ns
16	$t_{d(CKH\text{-}SDA10)}$	Delay time, CLKOUT1 high to SDA10 in high impedance mode/driven			5[†]	ns
17	$t_{d(CKH\text{-}SDRAS)}$	Delay time, CLKOUT1 high to $\overline{\text{SDRAS}}$ in high impedance mode/driven			5[†]	ns
18	$t_{d(CKH\text{-}SDCAS)}$	Delay time, CLKOUT1 high to $\overline{\text{SDCAS}}$ in high impedance mode/driven			5[†]	ns
19	$t_{d(CKH\text{-}SDWE)}$	Delay time, CLKOUT1 high to $\overline{\text{SDWE}}$ in high impedance mode/driven			5[†]	ns
20	$t_{d(CKH\text{-}SDCKE)}$	Delay time, CLKOUT1 high to SDA10 low/driven			5[†]	ns
21	$t_{d(CKH\text{-}SHD)}$	Delay time, CLKOUT1 high to HD in high impedance mode/driven		0	5[†]	ns
22	$t_{d(CKH\text{-}SHACK)}$	Delay time, CLKOUT1 high to HACK in low/driven			5[†]	ns
23	$t_{d(RESET\text{-}TRI)}$	Delay time, RESET low to signals low (or driven low-HACK)			2	cycles
24	$t_{d(RESET\text{-}DRV)}$	Delay time, RESET high to signals driven		2		cycles

[†] Values derived from characterization data and not tested

[‡] Values specified by design and not tested

NOTE 4: The $\overline{\text{RESET}}$ signal is not connected internally to the clock PLL circuit. The PLL, however, can need up to 210 µs to stabilize following device power up or after PLL configuration has been changed. During that time, $\overline{\text{RESET}}$ must be asserted to ensure proper device operation. See the PLL section for PLL lock times.

PRODUCT PREVIEW

PARAMETER MEASUREMENT INFORMATION

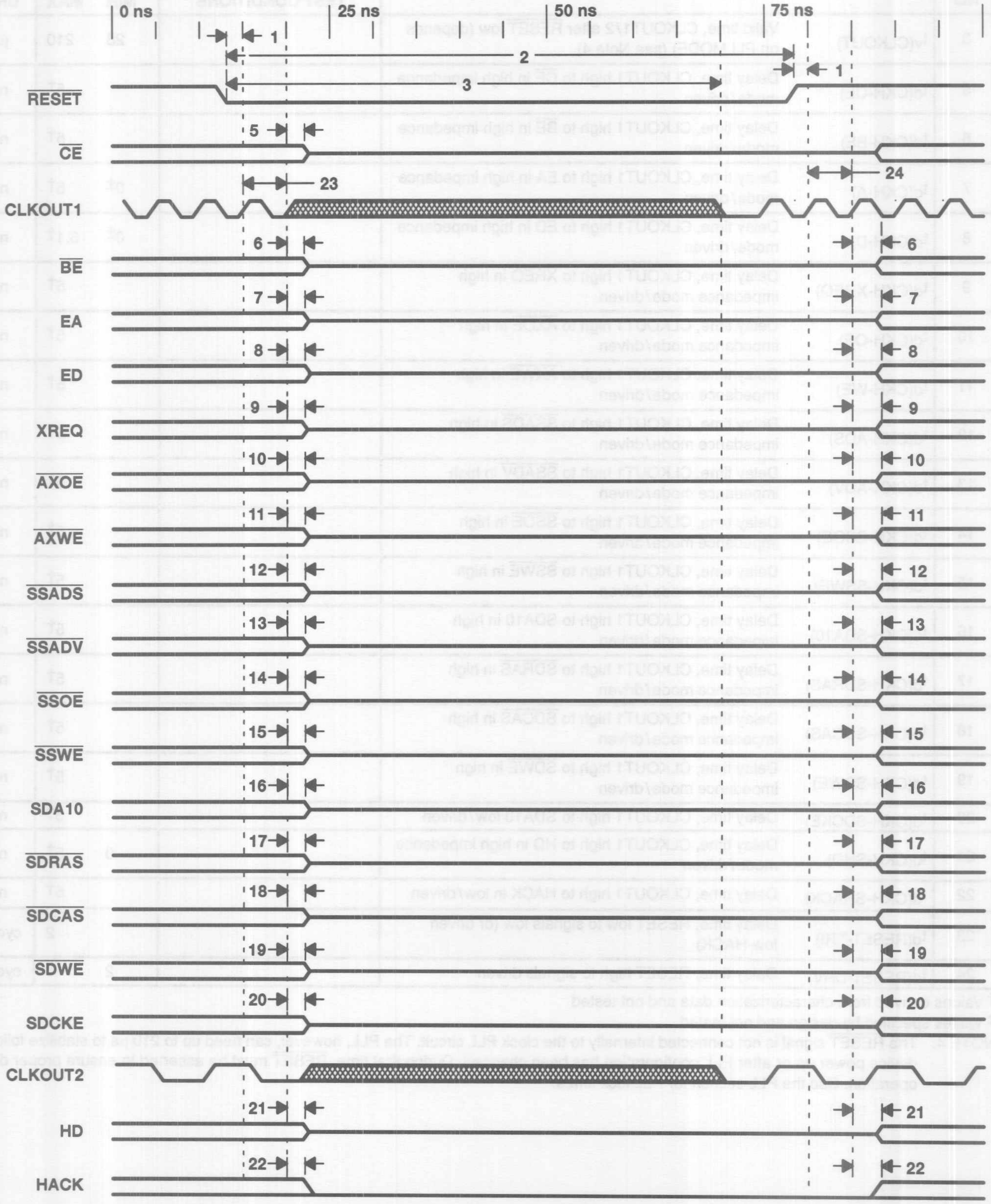


Figure 41. Reset-Cycle Timing

hold / hold-acknowledge timing

HOLD/HOLDA cycles are used by the host to disconnect the '320C6201 DSP from the system electrically so the host processor can access system resources directly. As shown on Figure 42, HOLD/HOLDA cycles put all output signals on the EMIF in the high-impedance mode (except HOLDA). To request the bus, the host processor drives the HOLD signal low. When the '320C6201 is ready, it responds by putting the EMIF bus in the high-impedance mode and asserting the HOLDA signal to announce to the host that the bus is no longer driven. The host has permission to use the EMIF bus as long as both HOLD and HOLDA control signals are low. Once the HOLD signal is de-asserted, the DSP responds by bringing the HOLDA signal high, at which point it resumes driving the EMIF outputs. Note that the HOLD/HOLDA logic is not affected by the state of the RESET signal.

timing requirements for the hold / hold-acknowledge cycle

NO			MIN	MAX	UNIT
1	$t_{su(CKH-HOLD)}$	Setup time, HOLD valid before CLKOUT1 high	1.5		ns

PRODUCT PREVIEW

hold/hold-acknowledge timing (continued)

switching characteristics for the hold/hold-acknowledge cycle

NO	PARAMETER		TEST CONDITIONS	MIN	MAX	UNIT
2	$t_{r(HOLDA)}$	Response time, $\overline{\text{HOLD}}$ high to $\overline{\text{HOLDA}}$ high		3		cycles
3	$t_{d(BUS)}$	Delay time, $\overline{\text{HOLDA}}$ low to EMIF and HPI buses driven			1.5	cycles
4	$t_{d(HOLDA)}$	Delay time, CLKOUT1 high to $\overline{\text{HOLDA}}$ valid			3.1	ns
5	$t_{d(CKH-CE)}$	Delay time, CLKOUT1 high to $\overline{\text{CE}}$ in the high-impedance mode/driven			3.1†	ns
6	$t_{d(CKH-BE)}$	Delay time, CLKOUT1 high to $\overline{\text{BE}}$ in the high-impedance mode/driven			3.1†	ns
7	$t_{d(CKH-A)}$	Delay time, CLKOUT1 high to EA in the high-impedance mode/driven		0‡	3.1†	ns
8	$t_{d(CKH-D)}$	Delay time, CLKOUT1 high to ED in the high-impedance mode/driven		0‡	4.5†	ns
9	$t_{d(CKH-XREQ)}$	Delay time, CLKOUT1 high to $\overline{\text{XREQ}}$ in the high-impedance mode/driven			3.1†	ns
10	$t_{d(CKH-OE)}$	Delay time, CLKOUT1 high to $\overline{\text{AXOE}}$ in the high-impedance mode/driven			3.1†	ns
11	$t_{d(CKH-WE)}$	Delay time, CLKOUT1 high to $\overline{\text{AXWE}}$ in the high-impedance mode/driven			3.1†	ns
12	$t_{d(CKH-ADS)}$	Delay time, CLKOUT1 high to $\overline{\text{SSADS}}$ in the high-impedance mode/driven			3.1†	ns
13	$t_{d(CKH-ADV)}$	Delay time, CLKOUT1 high to $\overline{\text{SSADV}}$ in the high-impedance mode/driven			3.1†	ns
14	$t_{d(CKH-SSOE)}$	Delay time, CLKOUT1 high to $\overline{\text{SSOE}}$ in the high-impedance mode/driven			3.1†	ns
15	$t_{d(CKH-SSWE)}$	Delay time, CLKOUT1 high to $\overline{\text{SSWE}}$ in the high-impedance mode/driven			3.1†	ns
16	$t_{d(CKH-SDA10)}$	Delay time, CLKOUT1 high to SDA10 in the high-impedance mode/driven			3.1†	ns
17	$t_{d(CKH-SDRAS)}$	Delay time, CLKOUT1 high to $\overline{\text{SDRAS}}$ in the high-impedance mode/driven			3.1†	ns
18	$t_{d(CKH-SDCAS)}$	Delay time, CLKOUT1 high to $\overline{\text{SDCAS}}$ in the high-impedance mode/driven			3.1†	ns
19	$t_{d(CKH-SDWE)}$	Delay time, CLKOUT1 high to $\overline{\text{SDWE}}$ in the high-impedance mode/driven			3.1†	ns
20	$t_{d(CKH-SDCKE)}$	Delay time, CLKOUT1 high to SDA10 in the high-impedance mode/driven			3.1†	ns

† Values derived from characterization data and not tested

‡ Values specified by design and not tested

PARAMETER MEASUREMENT INFORMATION

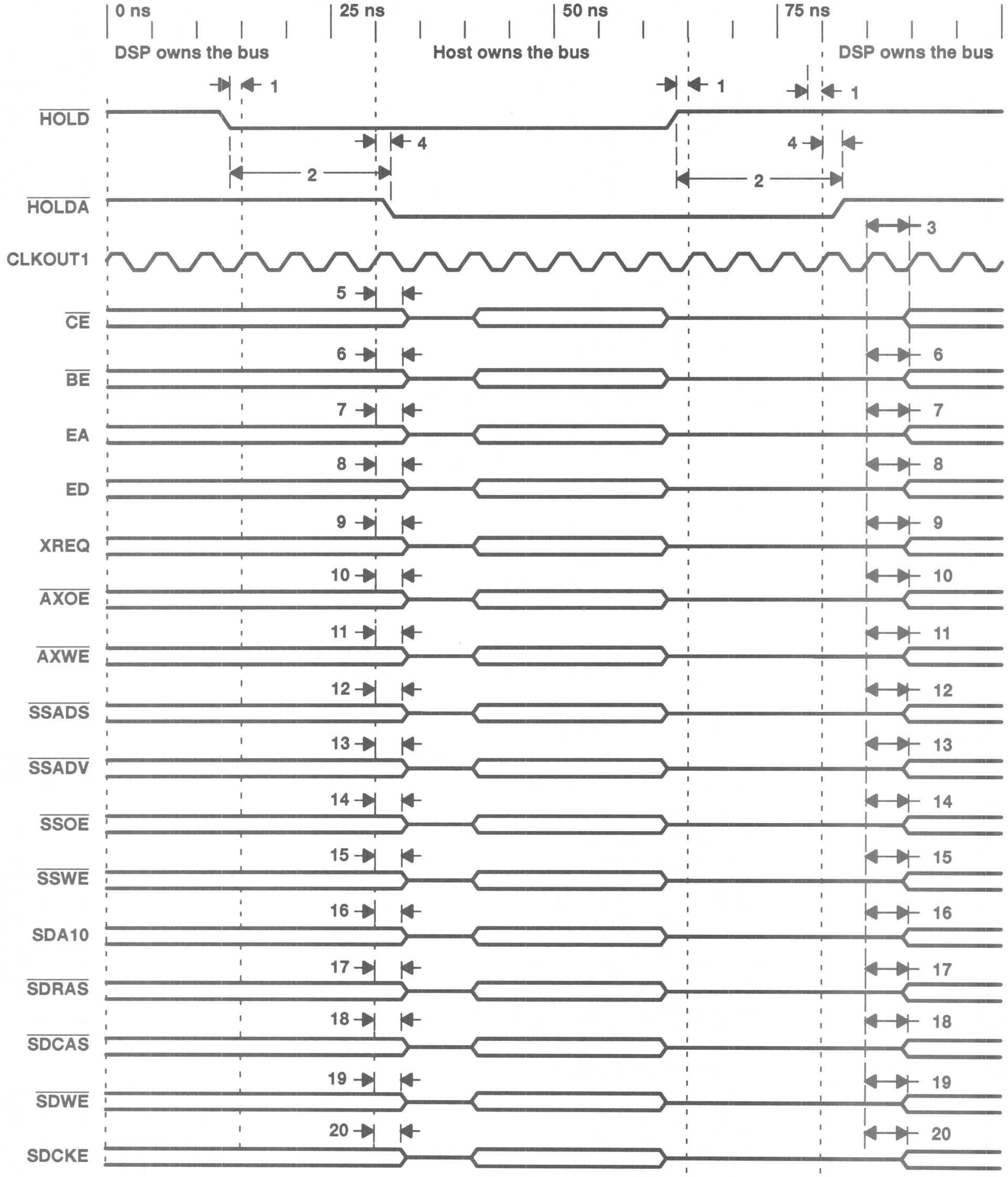


Figure 42. Hold / Hold-Acknowledge-Cycle Timing

PRODUCT PREVIEW

PRODUCT PREVIEW

host-port-interface cycles

Figure 43 shows signals to the HPI. Figure 44 and Figure 45 show read and write cycles using the 16-bit host-port interface. The HPI provides the host access to the internal data memory. To request access to the data memory, the host asserts the HREQ control signal. When the data memory controller (DMC) is ready to grant access to the host, it drives the HACK signal high. At that point, the host has control of the internal data memory to read and write to the '320C6201. When data transfer has completed, the host deasserts the HREQ signal. When the '320C6201 senses a change of state of the HREQ signal, it returns the HACK signal to the low state and regains control of the internal data memory. The timing of the handshaking signals HREQ and HACK is specified relative to the rising edge of CLKOUT1 and the rest of the host port timing parameters are specified in CLKOUT1 clock cycles.

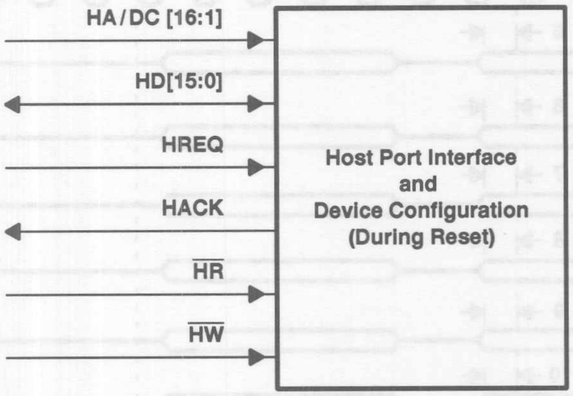


Figure 43. HPI and Device Configuration

timing requirements for the host-port cycles

NO			MIN	MAX	UNIT
1	$t_{su(HREQ\text{-}CKH)}$	Setup time, HREQ valid before CLKOUT1 high	1		ns
4	$t_{su(HA\text{-}HRW)}$	Set-up time, HA valid before HR/HW low	2		cycles
5	$t_{h(HA\text{-}HRW)}$	Hold time, HA valid after HR/HW high	0		cycles
6	$t_{w(HRW)}$	Pulse duration, width of HR and HW pulse	8		cycles
7	$t_{su(HDHIZ\text{-}HR)}$	Set-up time, HD high impedance (not driven by C6X) before HR low	0		cycles
11	$t_{su(HD\text{-}HW)}$	Set-up time, HD valid before HW low	2		cycles
12	$t_{h(HD\text{-}HW)}$	Hold time, HD valid after HW high	0		cycles
13	$t_{p(HRW)}$	Period between consecutive read and write cycles	12		cycles

switching characteristics for host-port cycles

NO			TEST CONDITIONS	MIN	MAX	UNIT
2	$t_{d(CKH\text{-}HACK)}$	Delay time, CLKOUT1 high to HACK valid			1	ns
3	$t_{r(HACK)}$	Response time, HREQ high to HACK high		7		cycles
8	$t_{d(HR\text{-}HDHIZ)}$	Delay time, HD high impedance after HR high		1†	2	cycles
9	$t_{d(HRL\text{-}HD)}$	Delay time, HR low to HD valid			6	cycles
10	$t_{v(HRH\text{-}HD)}$	Valid time, HD after HR high		1†	2	cycles

† Values specified by design and not tested

PARAMETER MEASUREMENT INFORMATION

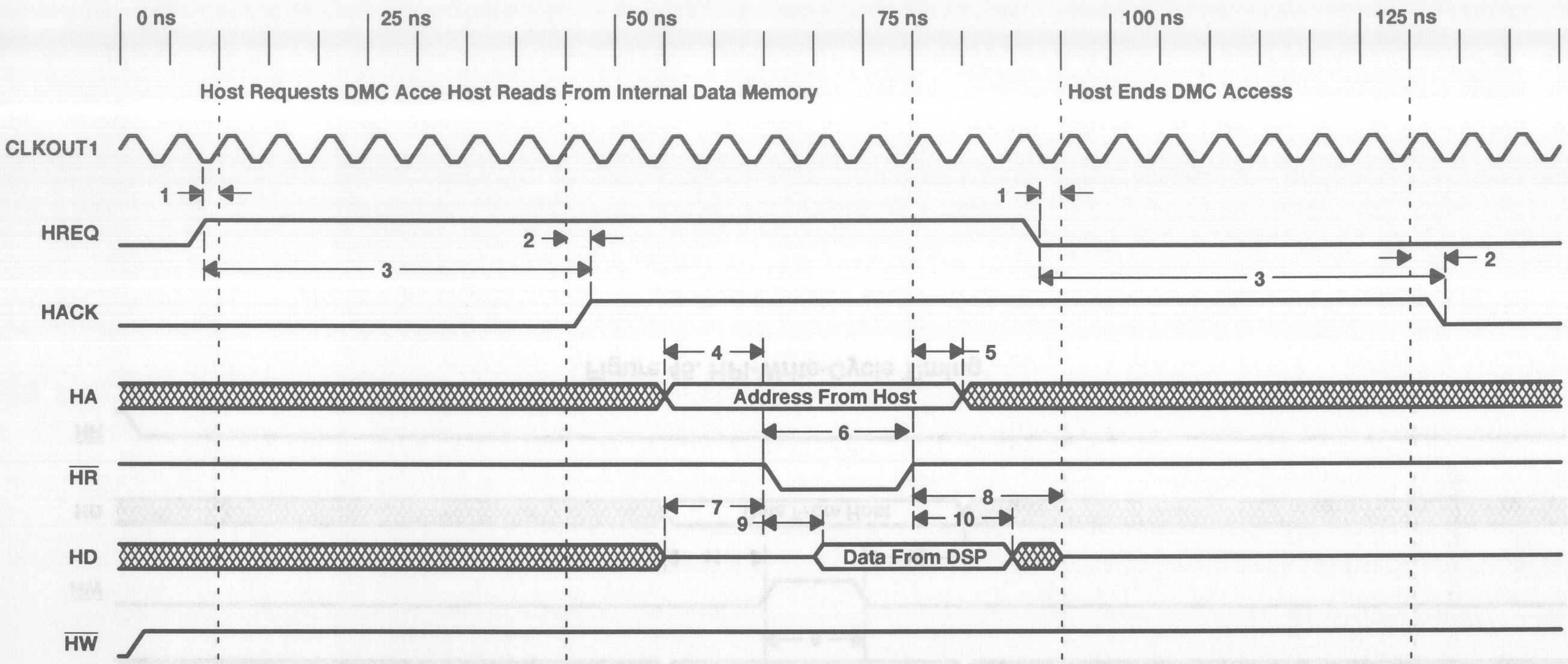


Figure 44. HPI-Read-Cycle Timing

PARAMETER MEASUREMENT INFORMATION

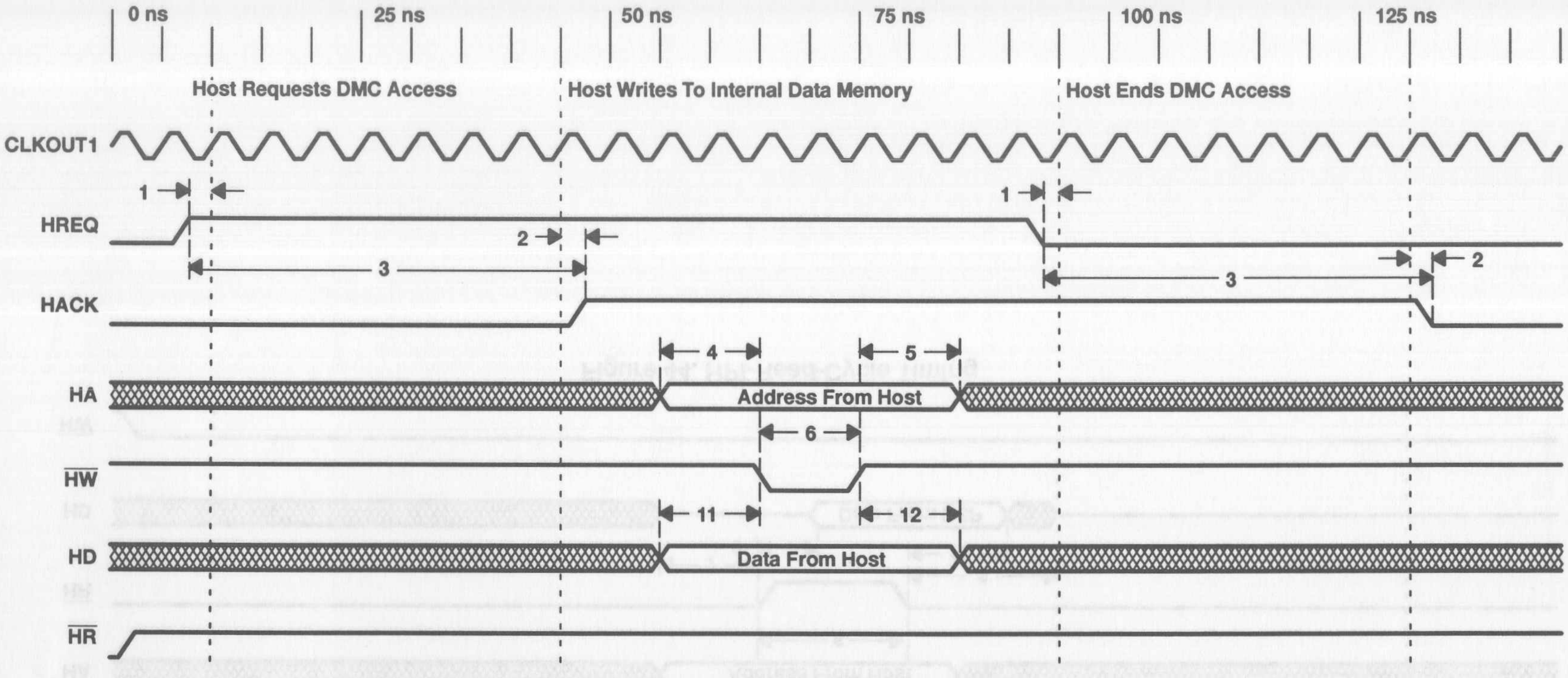
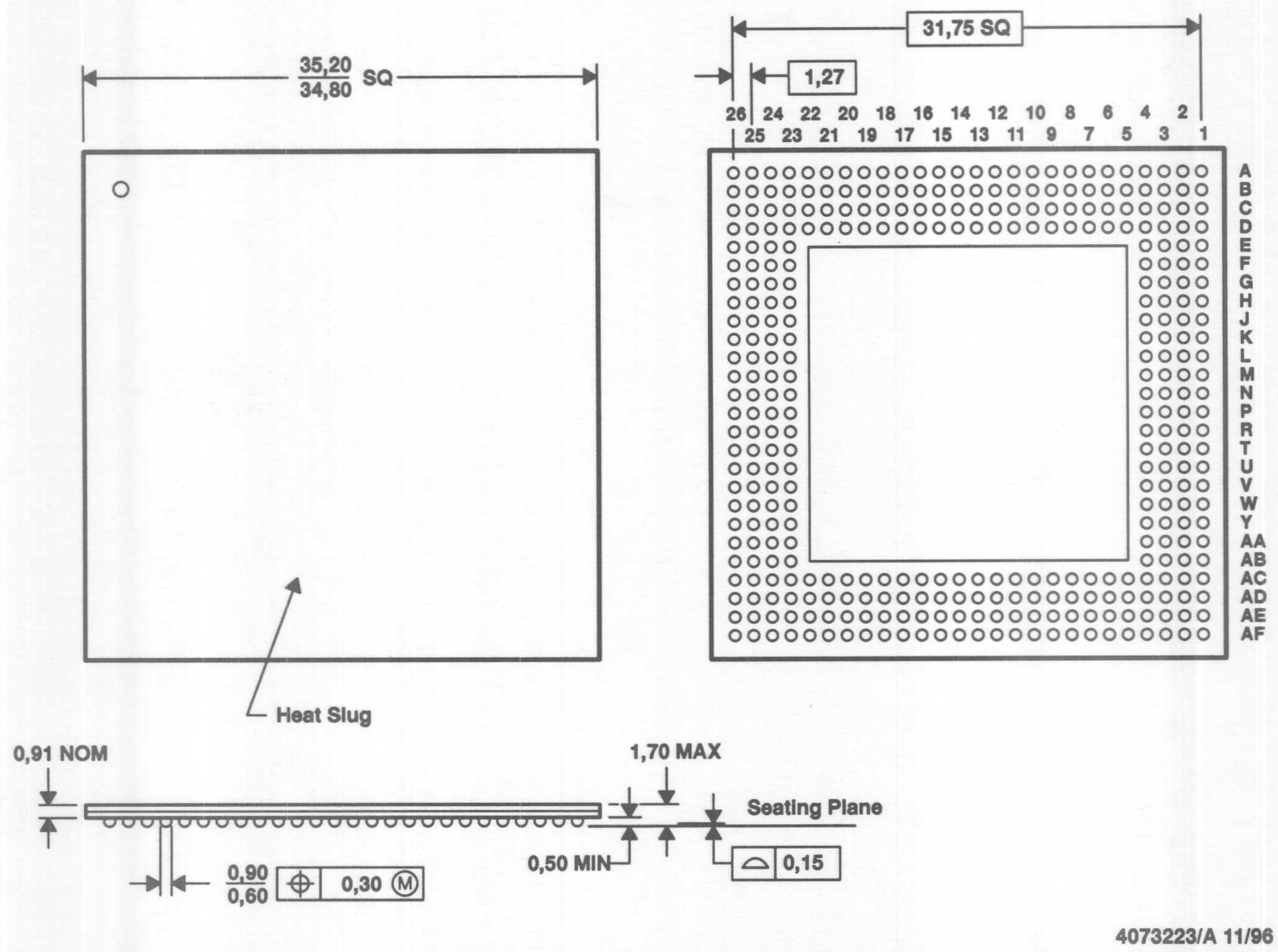


Figure 45. HPI-Write-Cycle Timing

MECHANICAL DATA

GGP (S-PBGA-N352) **PLASTIC BALL GRID ARRAY (CAVITY DOWN) PACKAGE**



NOTES: A. All linear dimensions are in millimeters.
B. This drawing is subject to change without notice.
C. Thermally enhanced die down plastic package with top surface metal heat slug.

thermal resistance characteristics for TMX320C6201 (SBGA package, 4-layer board, power = 5 W)

NO			°C/W	AIR FLOW LFPM†
1	$R\Theta_{JC}$	Junction-to-case	0.36	N/A
2	$R\Theta_{JA}$	Junction-to-free-air	10.97	0
3	$R\Theta_{JA}$	Junction-to-free-air	9.30	100
4	$R\Theta_{JA}$	Junction-to-free-air	8.45	200
5	$R\Theta_{JA}$	Junction-to-free-air	6.92	500

† LFPM = Linear Feet Per Minute

NOTES